Light and Shading

EECS 442 – Jeong Joon Park

Winter 2024, University of Michigan

http://eecs442.github.io

Administrivia

- HW1 Due in a Week
- Please sign up for Piazza
- Piazza highlights:
  - student answers (please pay it forward!)
  - Would be great if you can participate in discussions!
  - TAs are very hard-working! Constantly on Piazza.
- Collaborations:
  - How many of you have found study mates?
  - Highly encourage posting and participating to study buddy search on Piazza -- 16 groups so far

Thoughts on Homework

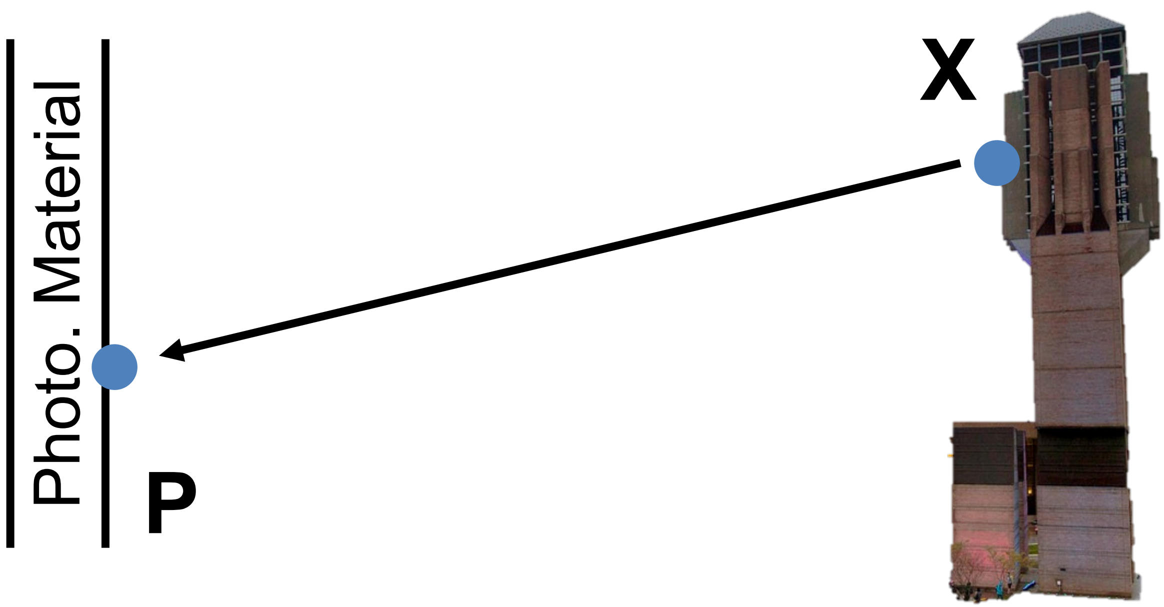
- Some self-study will be required for:
  - Reinforcing concepts
  - Learning libraries (there's great documentation!)
  - Often "self-study" parts won't be indicated
- Try first aid tips for debugging
  - Print ***X.shape***, X.dtype, X when you're stuck
  - plt.imsave("debug.png",X)
  - Read documentation
- This is a skill you need if you're going to do rewarding programming

Recap: Projection

$$\text{Image} \rightarrow \boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R}, \boldsymbol{t}]\boldsymbol{X} \leftarrow \text{World}$$

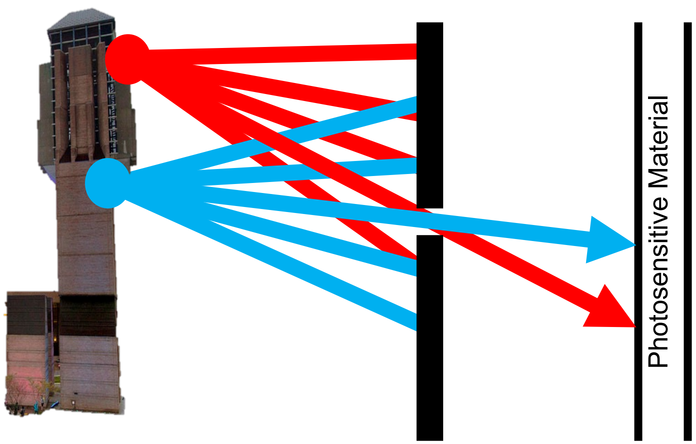
Intrinsic

Extrinsic



Recap: Lenses

Pinhole Model



Mathematically correct
Not quite correct in practice
Reasonable approximation

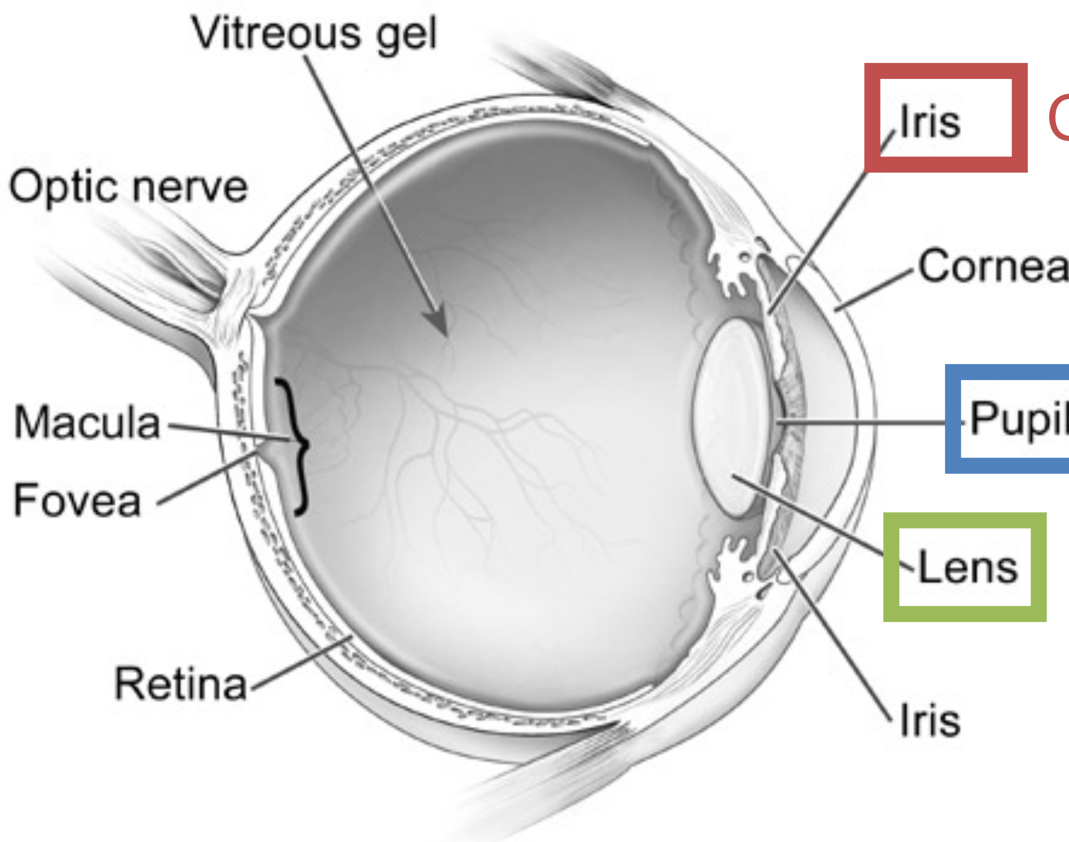
Reality: Lenses

Necessary in practice
Introduce complications
Complications fixable

Today

- A little bit about light and how you represent it
- A little bit about lighting and how it works

Your Very Own Camera



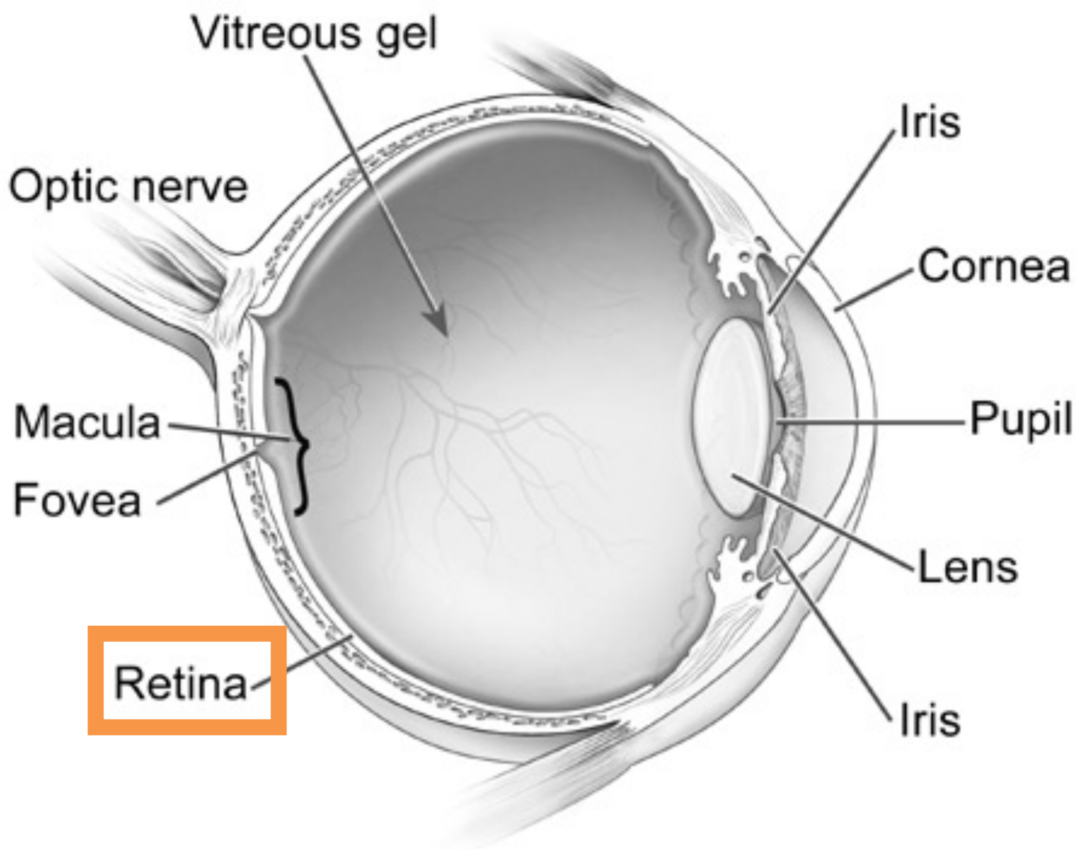
Controls pupil

Aperture letting in light

Lens…

Slide Credit: NIH

Your Very Own Camera



Where's the film/CCD?

Slide Credit: NIH

Demo Time

https://tinyurl.com/442w23

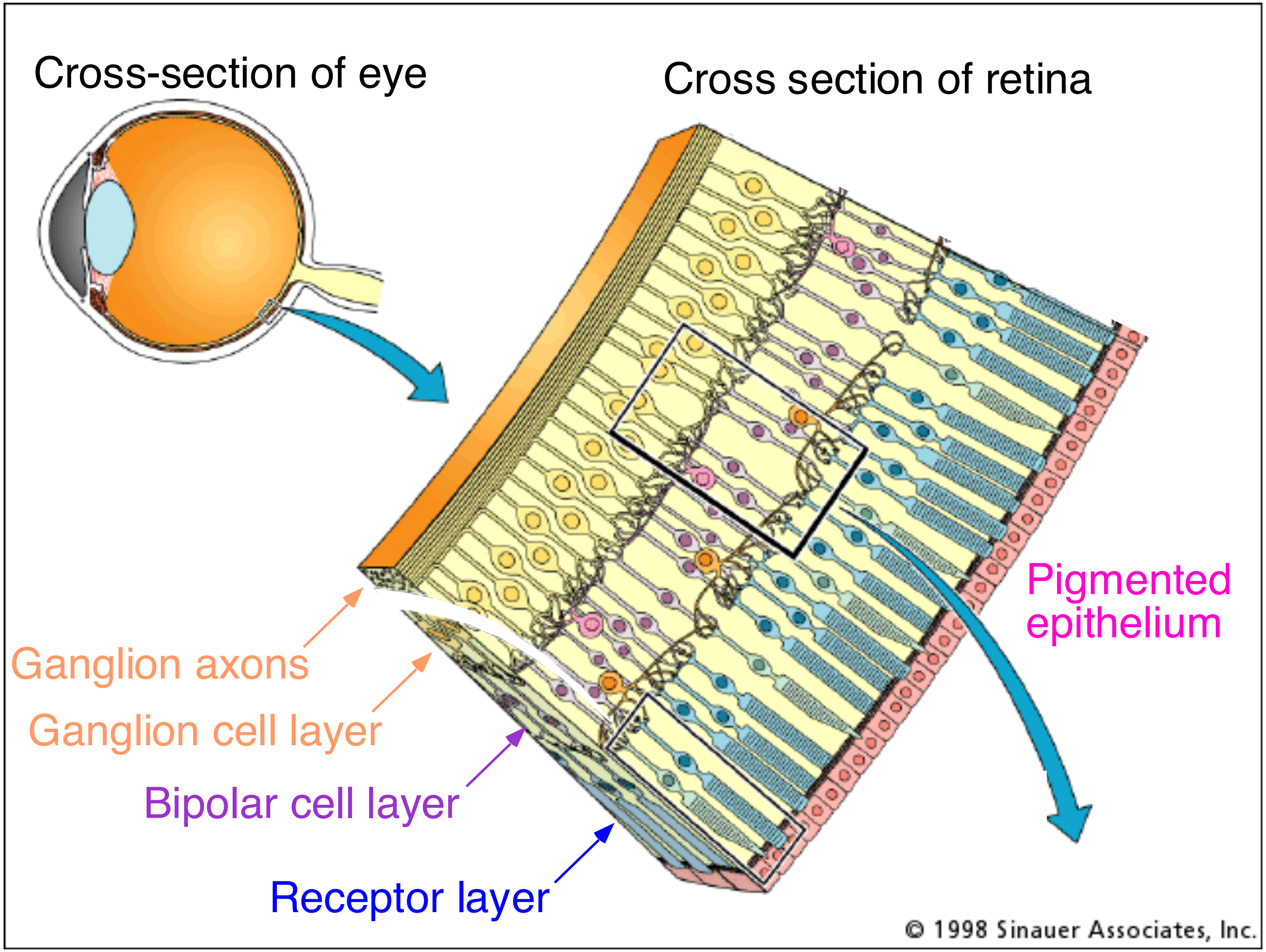
Made by David Fouhey

OHIO STATE

+

+

What is Retina/Film Made Of?



Slide Credit: J. Hays

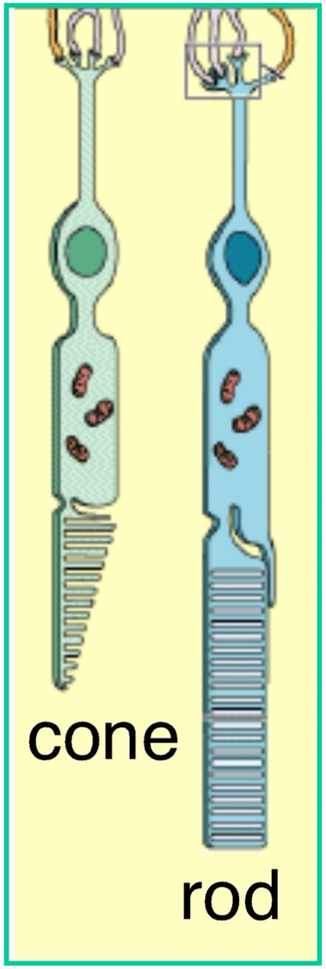
Two Type of Photo Receptors

Cones

cone-shaped
less sensitive
operate in high light
color vision

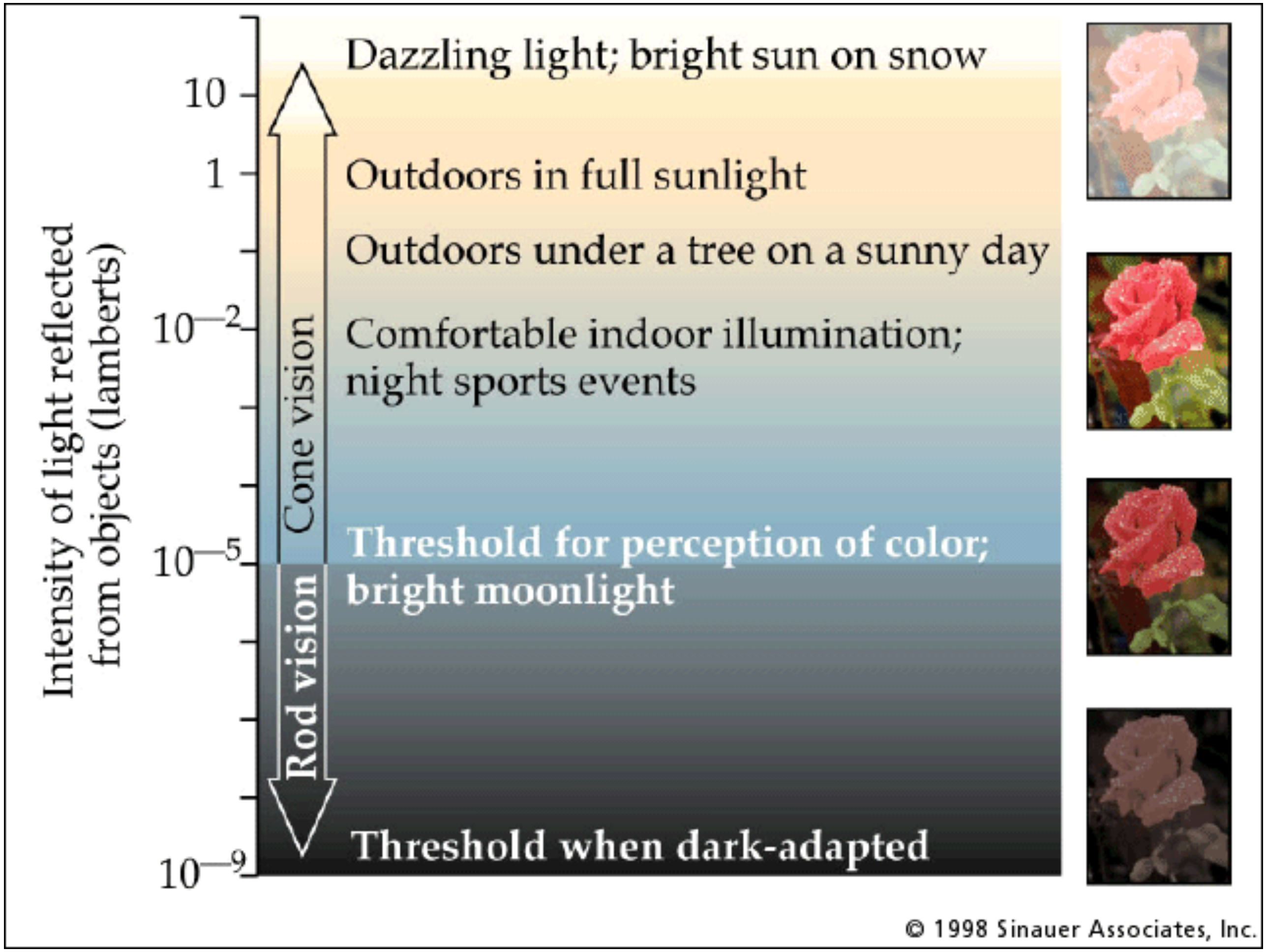
Rods

rod-shaped
highly sensitive
operate at night
gray-scale vision



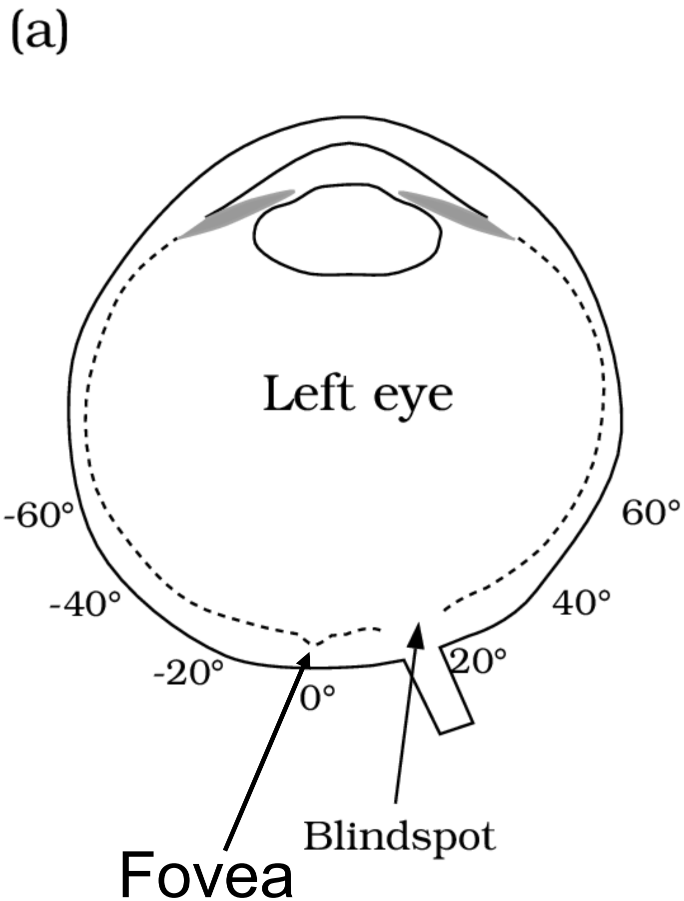
Slide Credit: J. Hays

Rod / Cone Sensitivity



Slide Credit: J. Hays

Rod/Cone Distribution



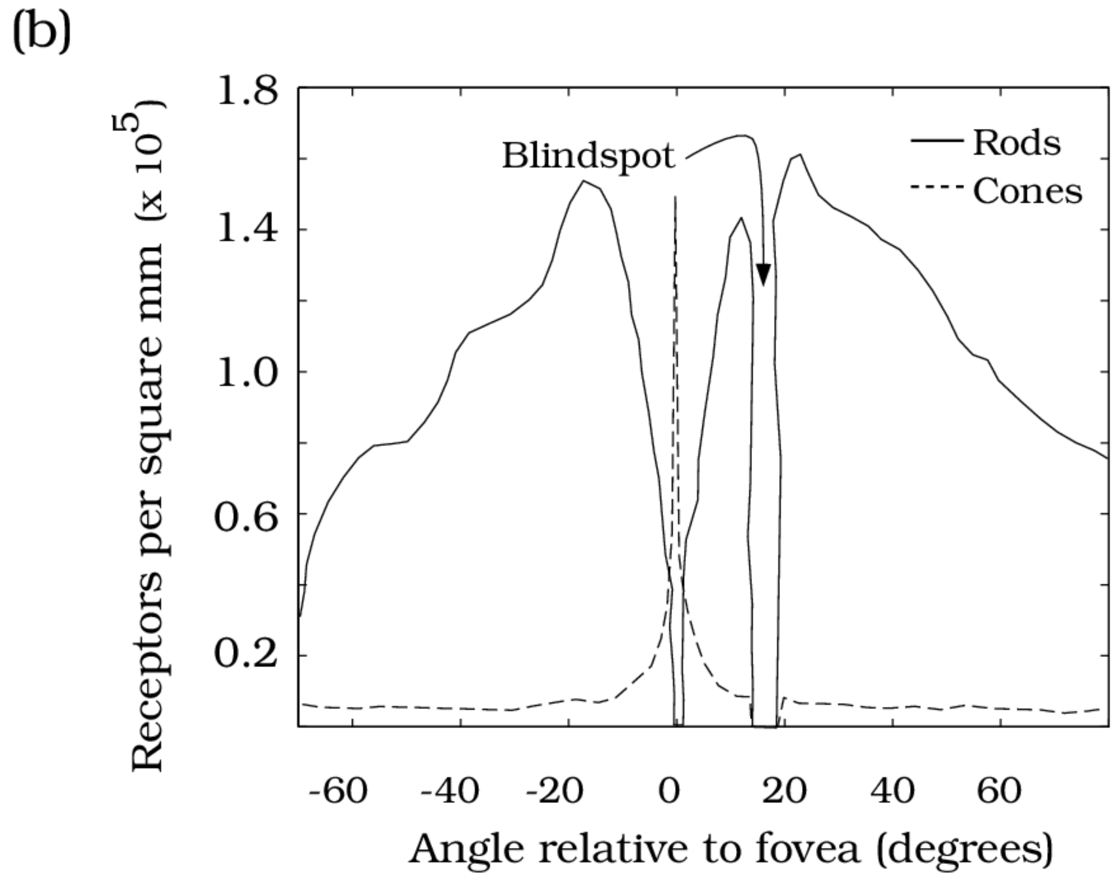
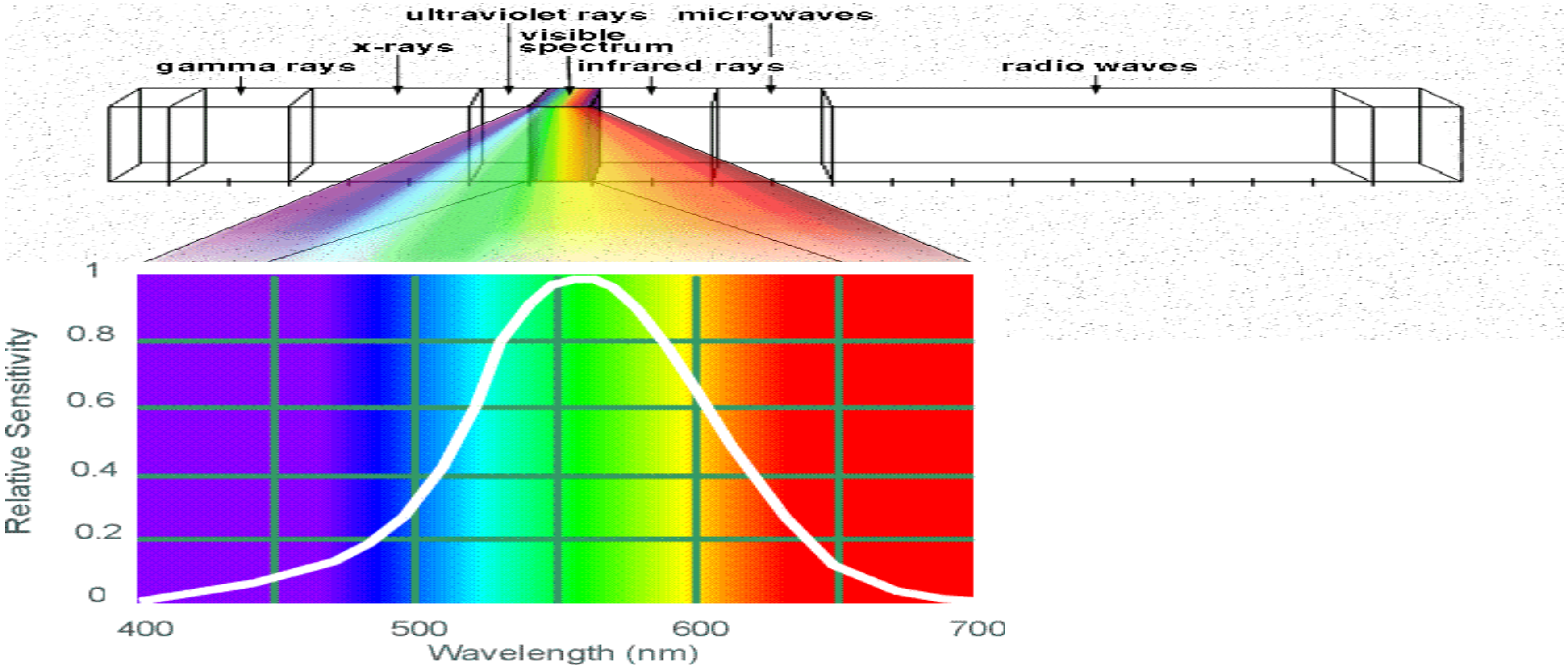


Diagram Credit: B. A. Wandell, *Foundations of Vision*

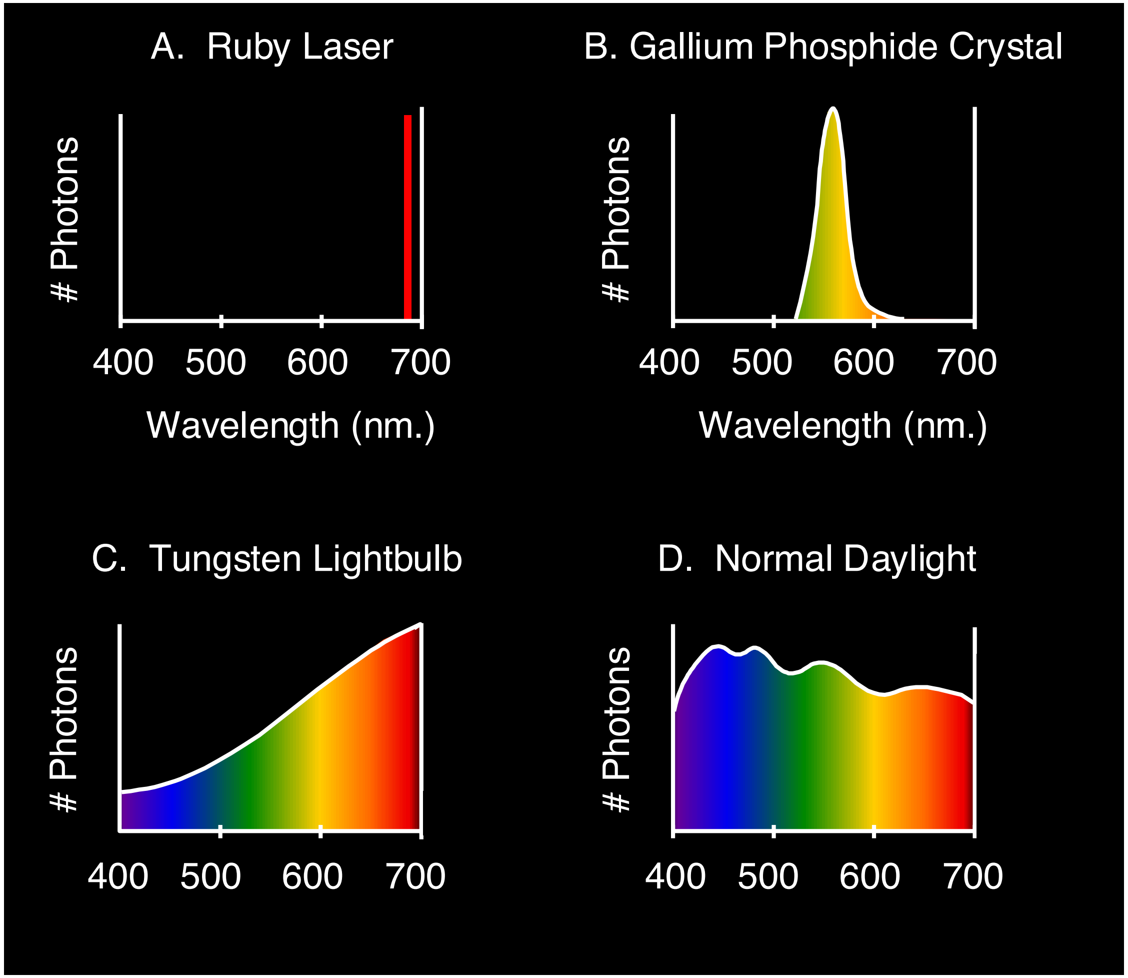
Electromagnetic Spectrum



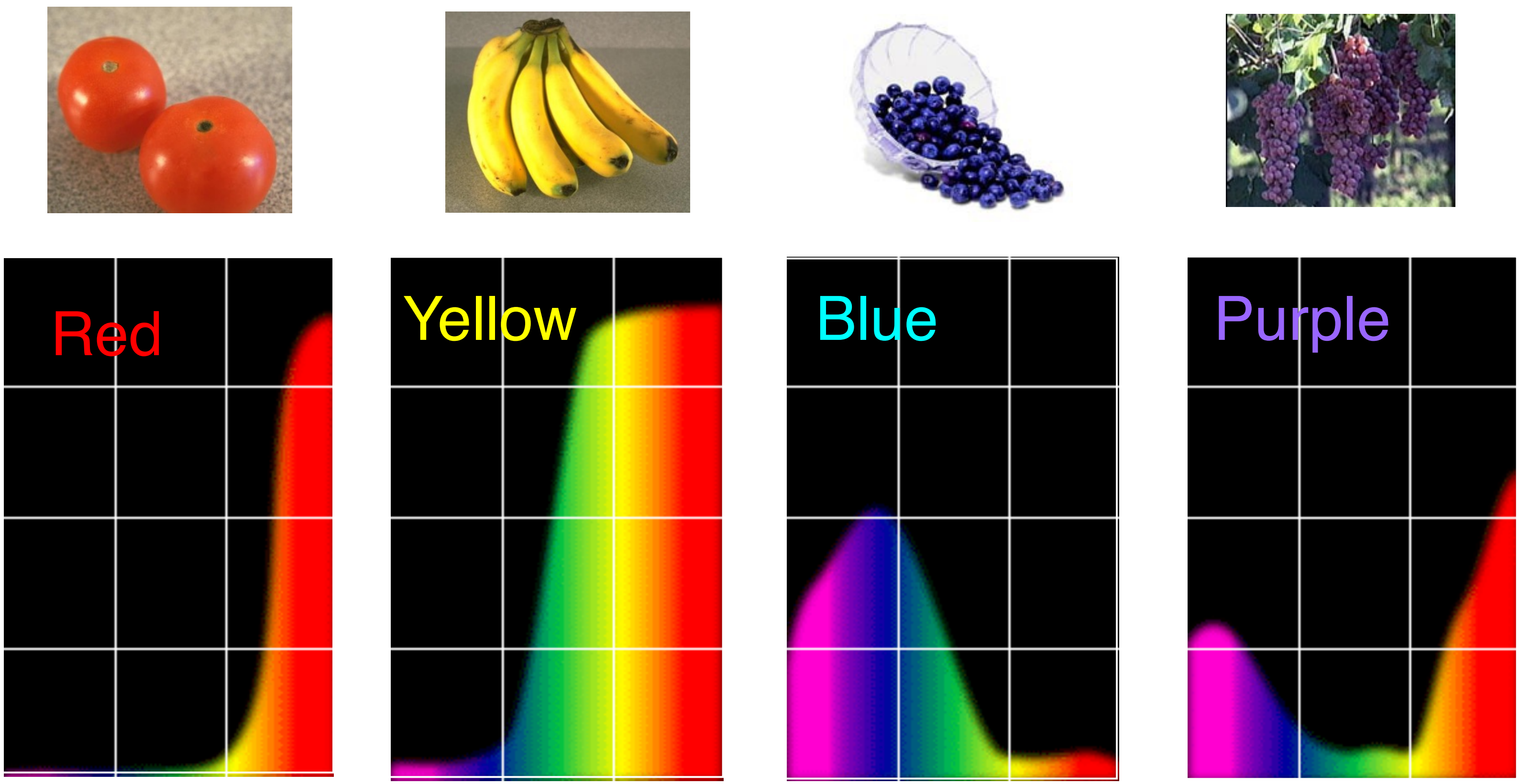
Why do we see light in these wavelengths?

Slide Credit: J. Hays

The Physics of Light

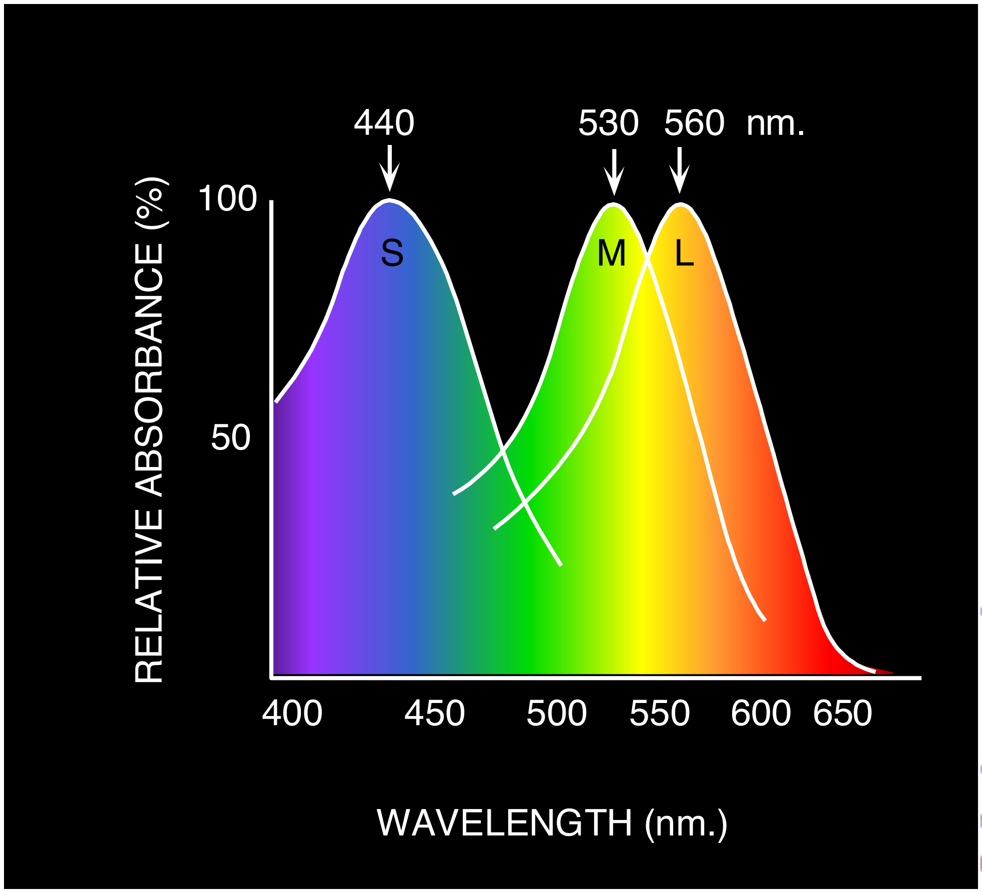


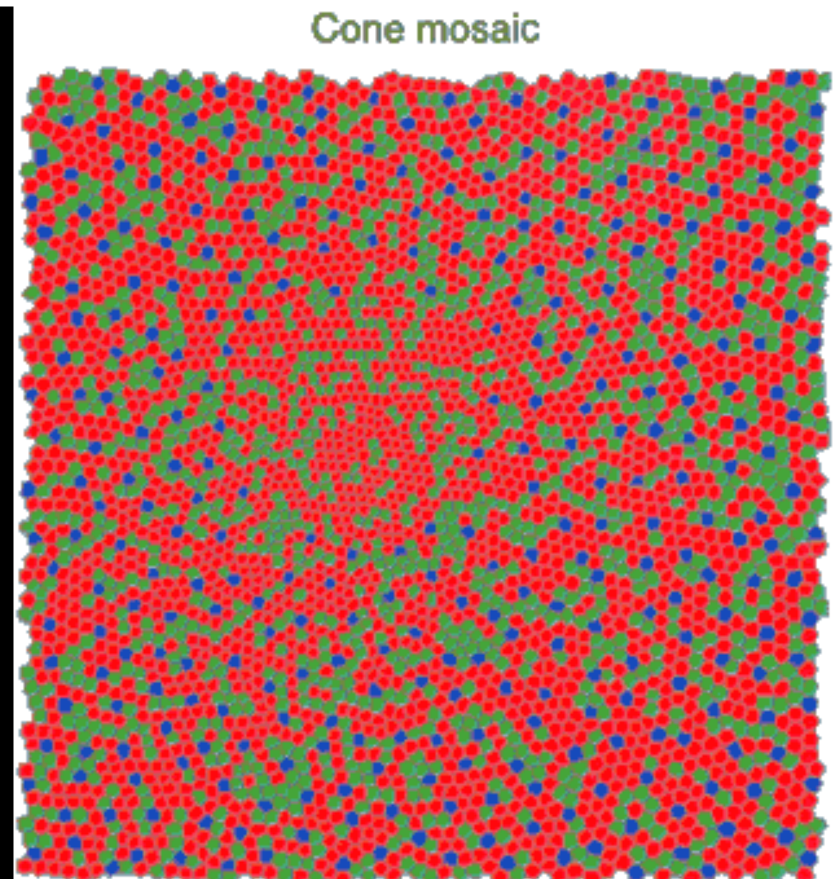
The Physics of Light



Slide Credit and Copyright: S. Palmer

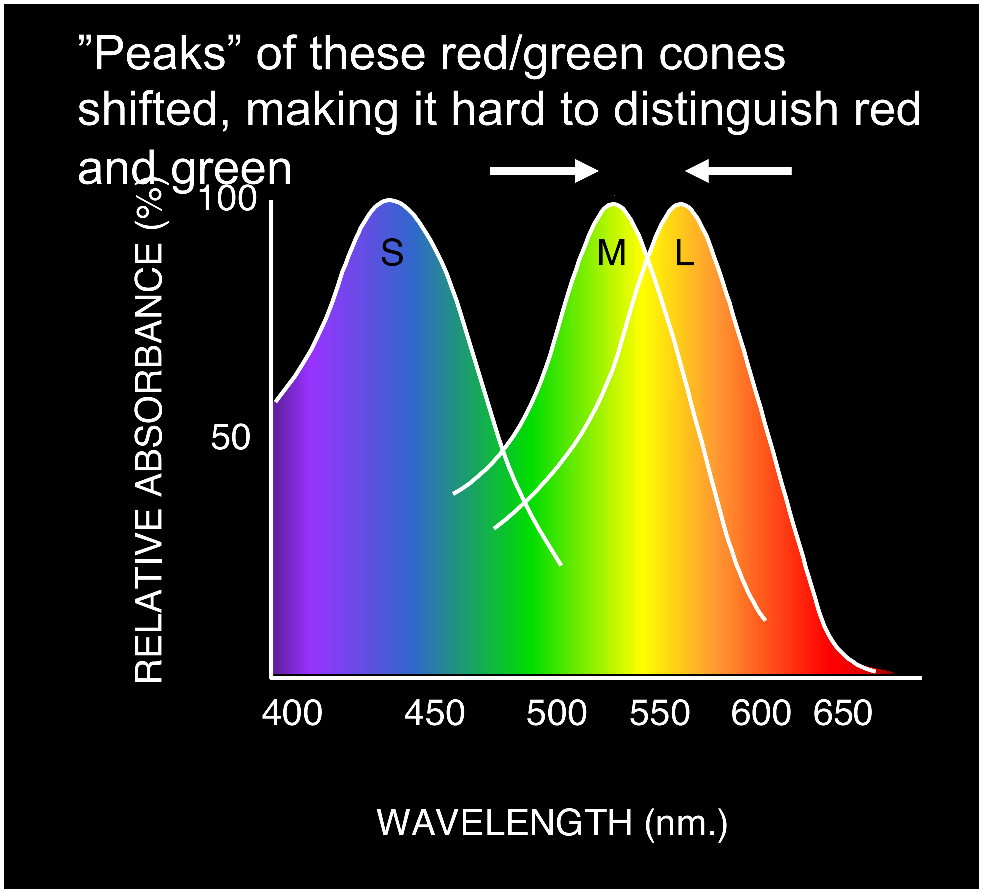
The Physics of Light





Slide Credit and Copyright: S. Palmer

Red-Green Color Blindness



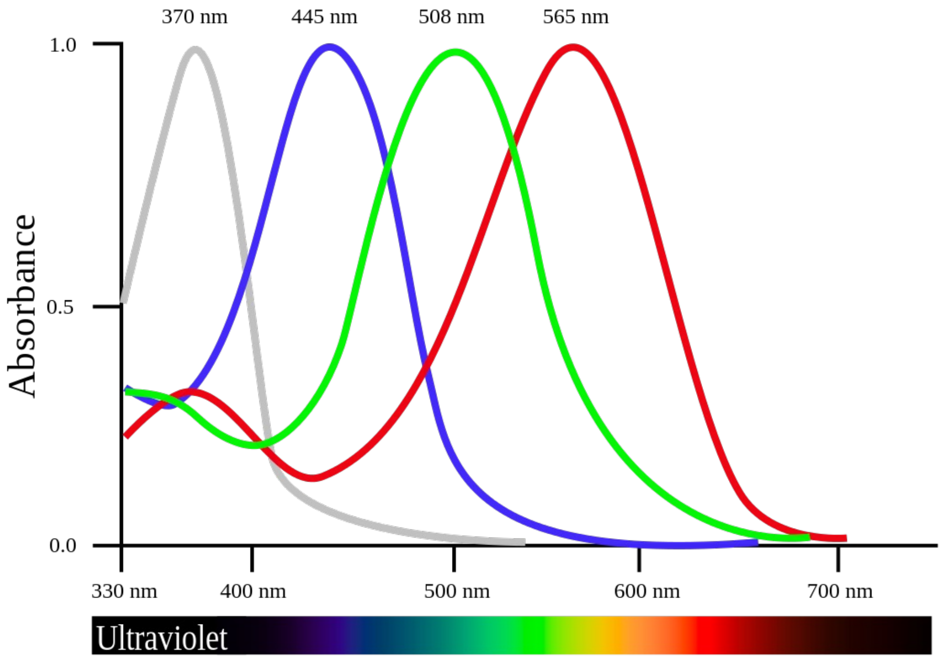
Four possibilities:

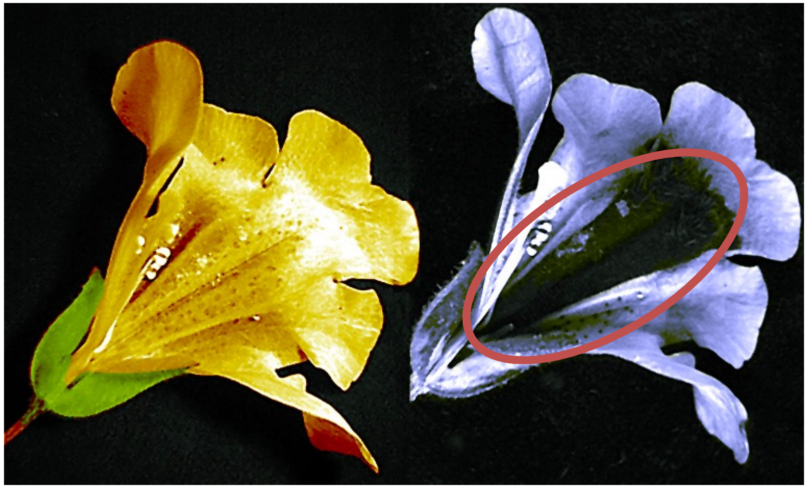
- **Deuteranomaly**: Green cone shifted toward red
- **Protanomaly**: Red cone shifted toward green
- **Deuteranopia**: Green cone missing
- **Protanopia**: Red cone missing

Color Vision in Animals

Birds have 4 cone types: can see ultraviolet light

Some flowers have "Nectar Guides" in UV





Visible Light UV light

Image source: [Wikipedia]

Image Source: [Wikipedia]

JANUARY 11, 2022 | ARTS & CULTURE

Could Claude Monet See Like a Bee?

A harrowing eye surgery may have given the impressionist painter the ability to see UV light.

by Sam Kean

Detail, *Water Lilies (Nymphéas)*, Claude Monet, ca. 1915–1926.

Carnegie Museum of Art/Wikimedia Commons

Color Vision in Animals

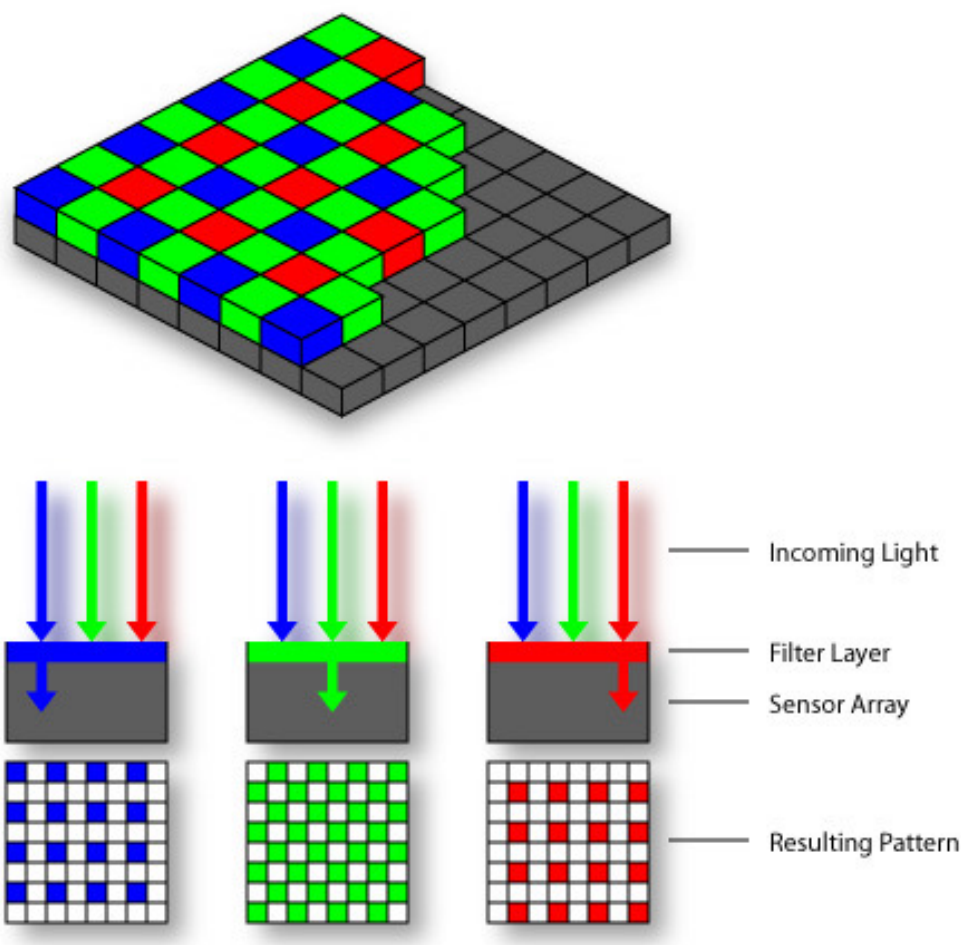
Mantis Shrimp: Up to 16 types of photoreceptors!
Can also detect polarization of light!

Image source: [Wikipedia]

How Do We Get Light?

- Cameras!

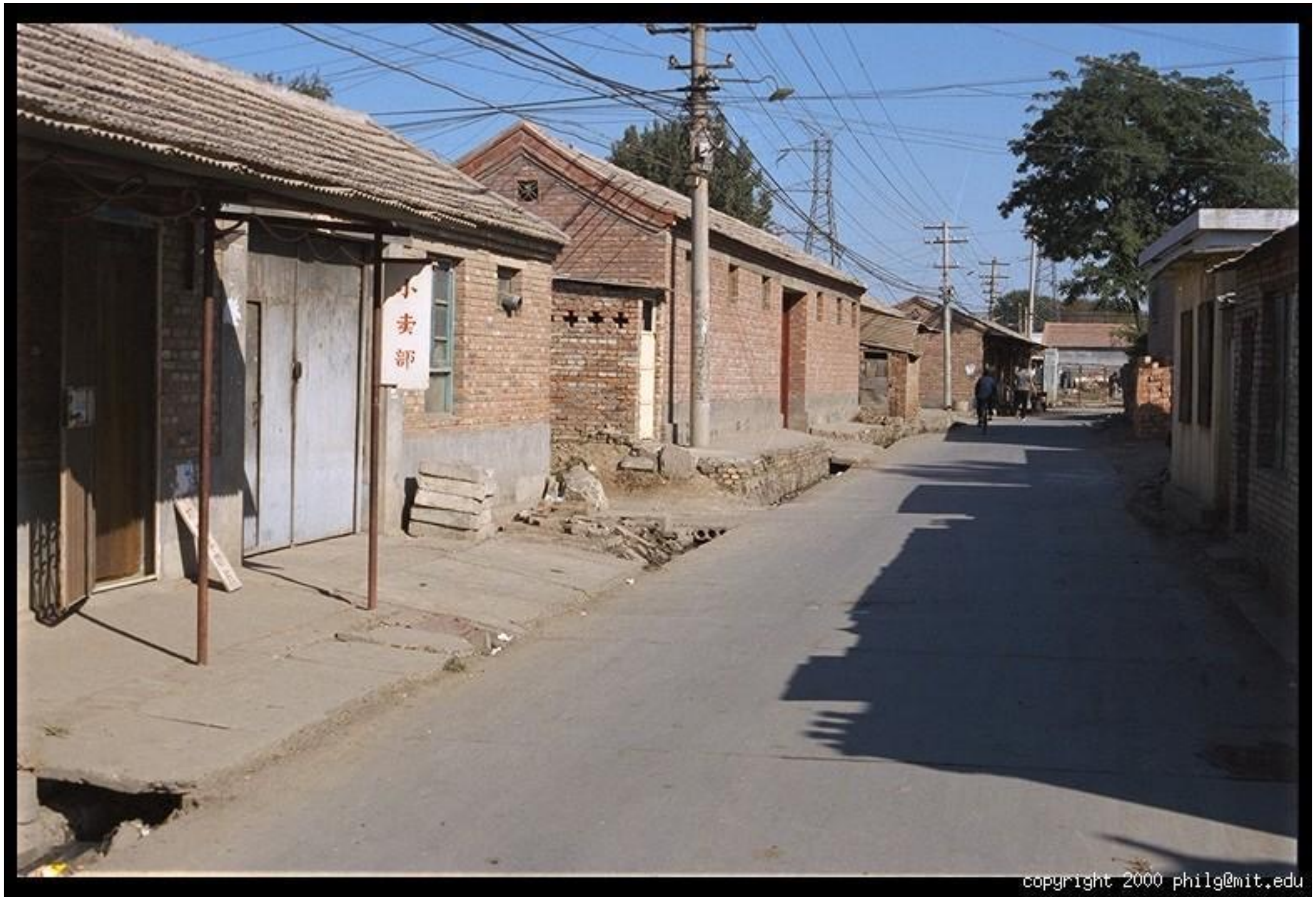
Artificial Cones



Estimate RGB at ‘G’ cells from neighboring values

Slide Credit: S. Seitz

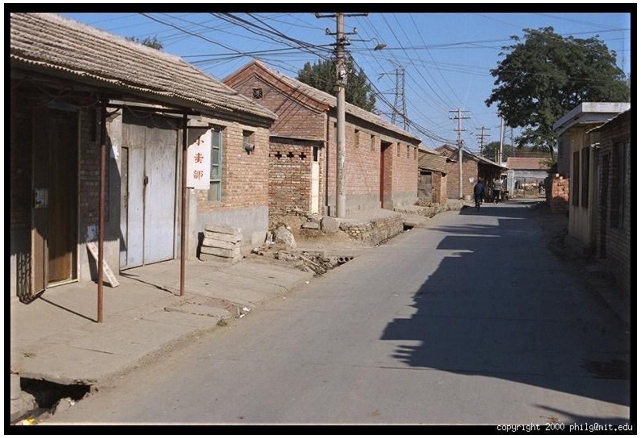
Color Image



Slide Credit: J. Hays

Color Image

Combined



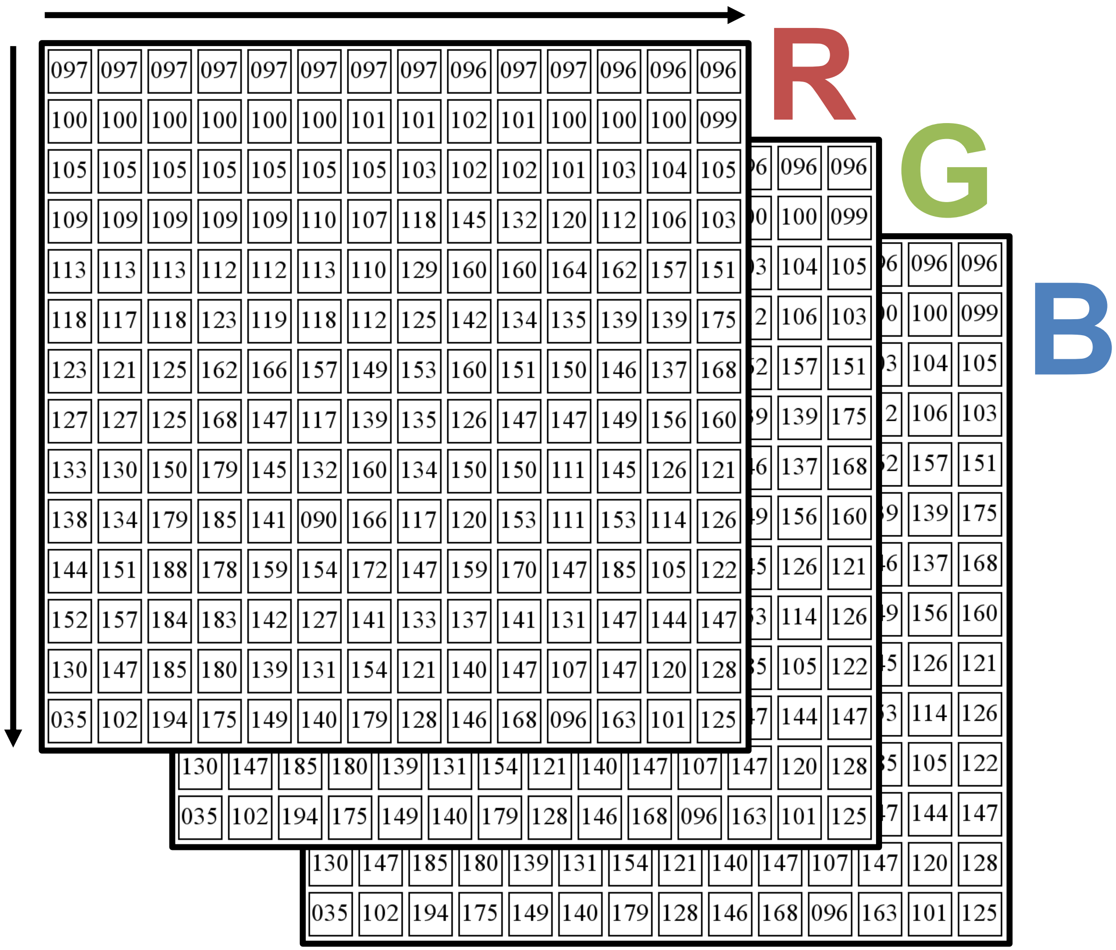
Red

Green

Blue

Slide Credit: J. Hays

Images in Python



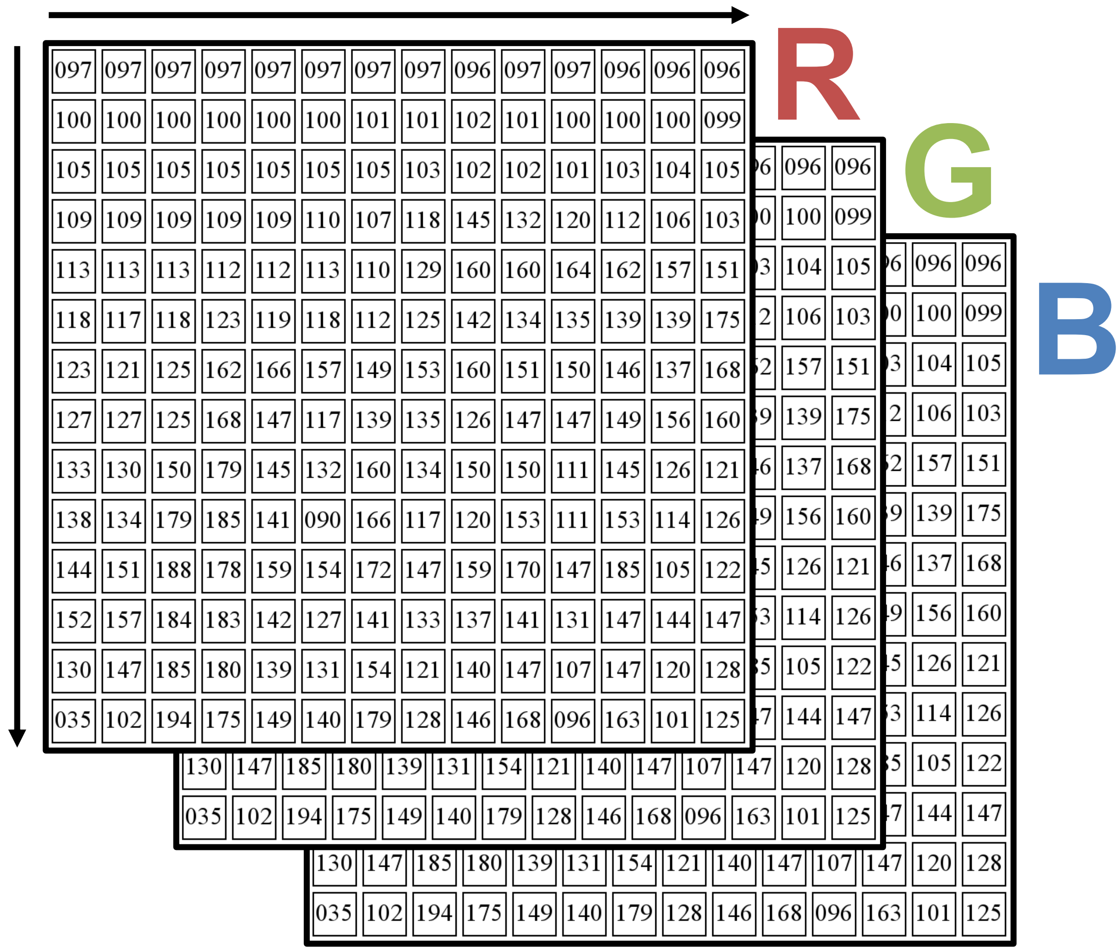
Images in Python

Images are matrix / tensor `im`

`im[0,0,0]`
  top, left, red

`im[y,x,c]`
  row y, column x, channel c

`im[H-1,W-1,2]`
  bottom right blue



Slide inspired by James Hays

Images in Python

Images are matrix /

`im[0,0,0]`

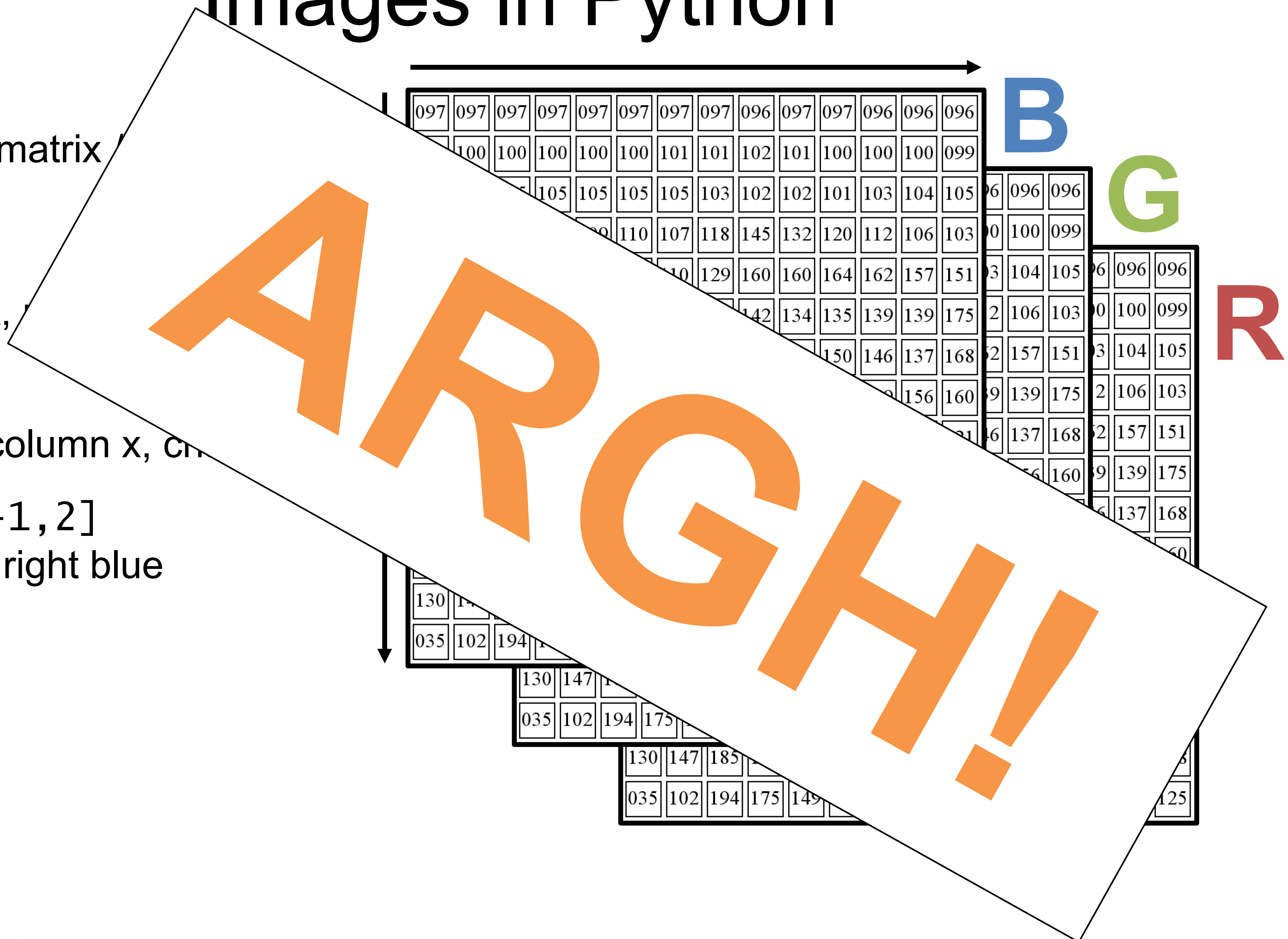
top, left, /

`im[y,x,c]`

row y, column x, ch

`im[H-1,W-1,2]`

bottom right blue

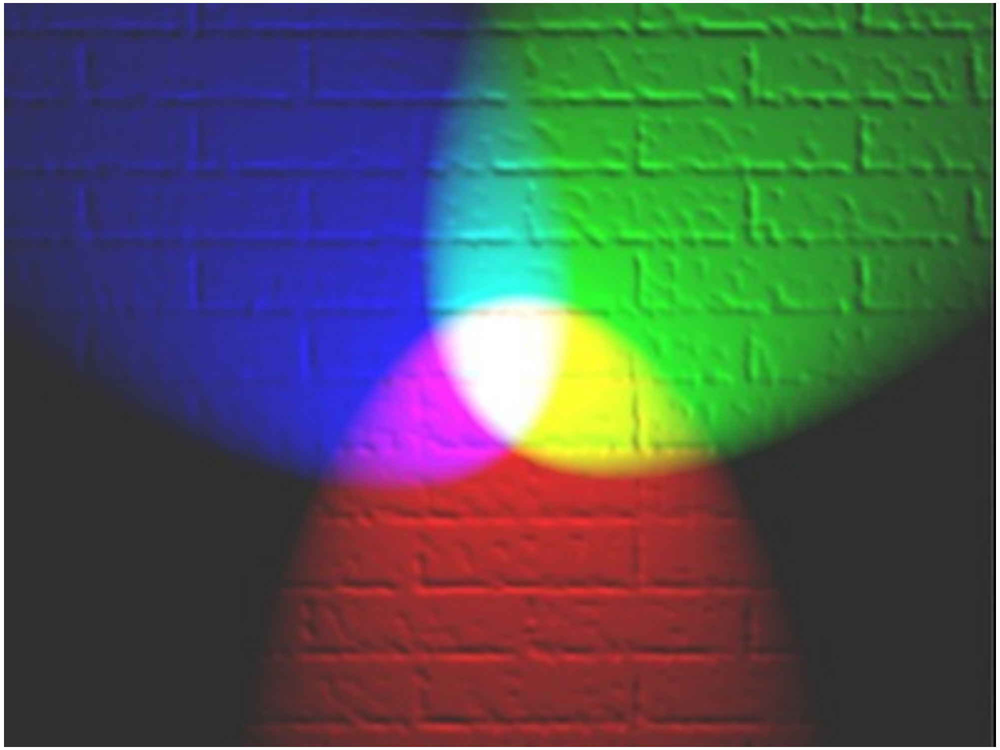


Slide inspired by James Hays

5 Things To Remember

1. Origin is top left
2. Rows are first index (**what's the fastest direction for accessing?)**
3. Usually referred to as Height x Width
4. Typically stored as uint8 [0,255]
5. for y in range(H): for x in range(W): will run <u>1 million times </u>for a 1000x1000 image. *A 4GHz processor can do only 4K clock cycles per pixel per second.*
6. *In PyTorch: Channel x Height x Width*

Representing Colored Light



Discussion time: how many numbers do you actually need for colored light? Assume all tuples (R,G,B) are legitimate colors (they are).

Image Credit: http://en.wikipedia.org/wiki/File:RGB_illumination.jpg

5 difference in G

3 difference in G

2 difference in G

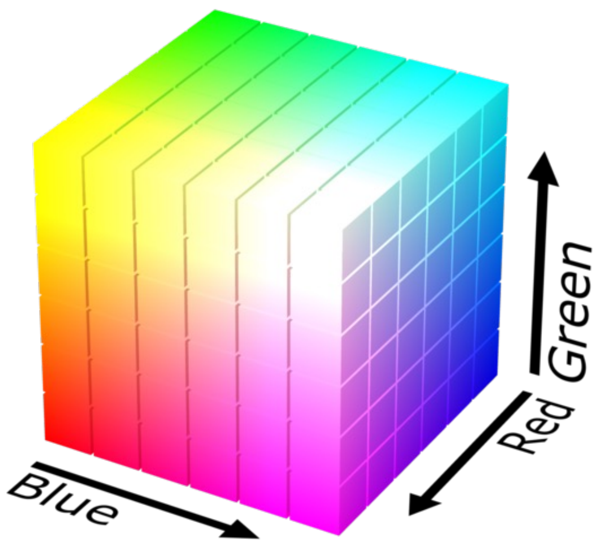
1 difference in G

5 difference in B

One Option: RGB

Pros
1. Simple
2. Common

Cons
1. Distances don't make sense
2. Correlated



R

G

B

Slide Credit: J. Hays, RGB cube: https://en.wikipedia.org/wiki/RGB_color_model

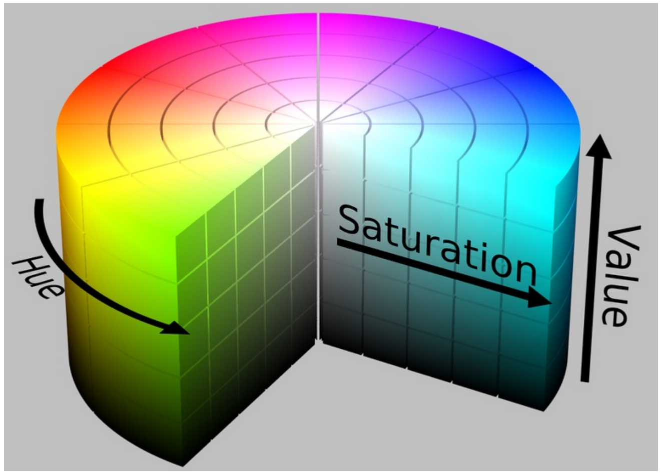
RGB

Photo credit: J. Hays

Another Option: HSV

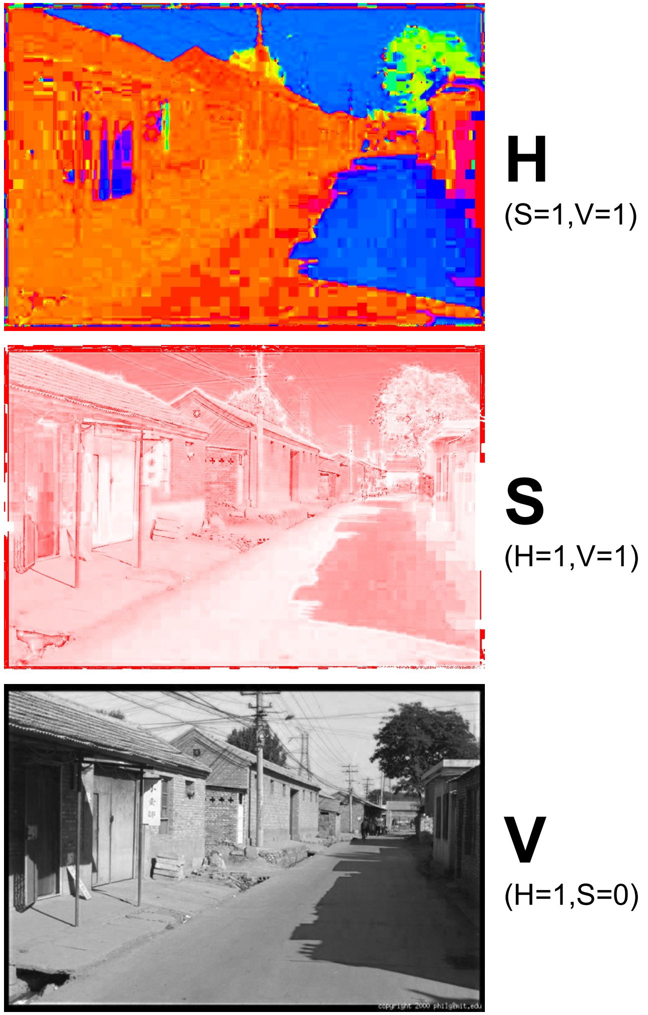
<u>Pros</u>

1. Intuitive for picking colors
2. Sort of common
3. Fast to convert



<u>Cons</u>

1. Perceptual Non-Linearity
2. Lack of Precision for Dark Colors



Slide Credit: J. Hays, HSV cylinder: https://en.wikipedia.org/wiki/HSL_and_HSV

HSV

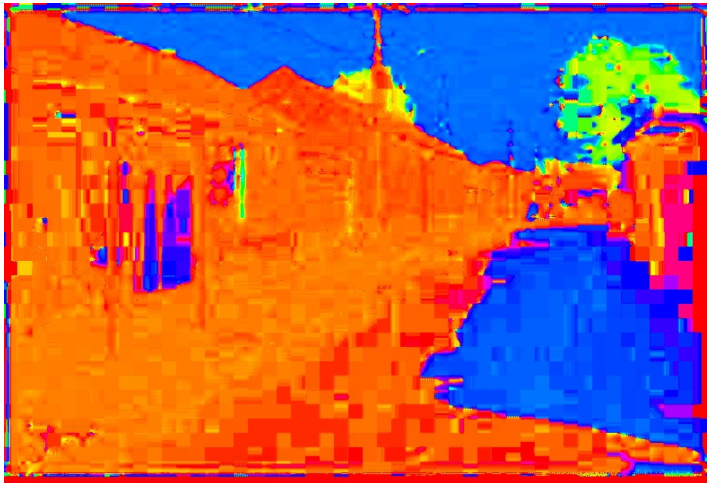


Photo credit: J. Hays

Another Option: YCbCr/YUV

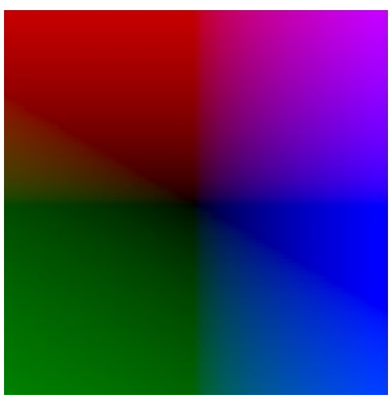
<u>Pros</u>

1. Great for transmission / compression

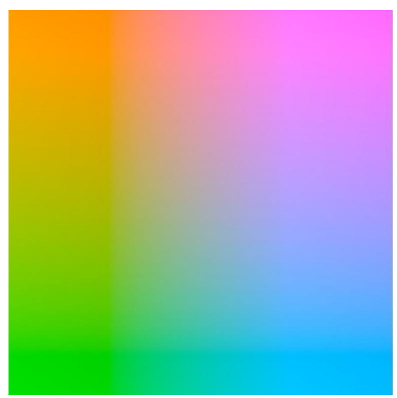
<u>Cons</u>

1. Not perceptual linear

Y = 0.1



Y = 0.5



Y
(Cb=0.5, Cr=0.5)

Cb
(Y=0.5, Cr=0.5)

Cr
(Y=0.5, Cb=05)

Slide Credit: J. Hays, YUV cube: https://en.wikipedia.org/wiki/YUV

YCbCr

Photo credit: J. Hays

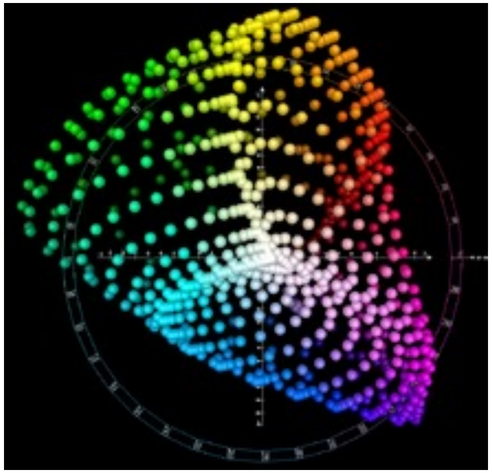
Another Option: Lab

<u>Pros</u>

1. Distances correspond with human judgment
2. Safe

<u>Cons</u>

1. Complex to calculate (don't write it yourself, lots of fp calculations)



L
(a=0,b=0)

a
(L=65,b=0)

b
(L=65,a=0)

Slide Credit: J. Hays, Lab diagram cube: https://en.wikipedia.org/wiki/CIELAB_color_space

Lab

Photo credit: J. Hays

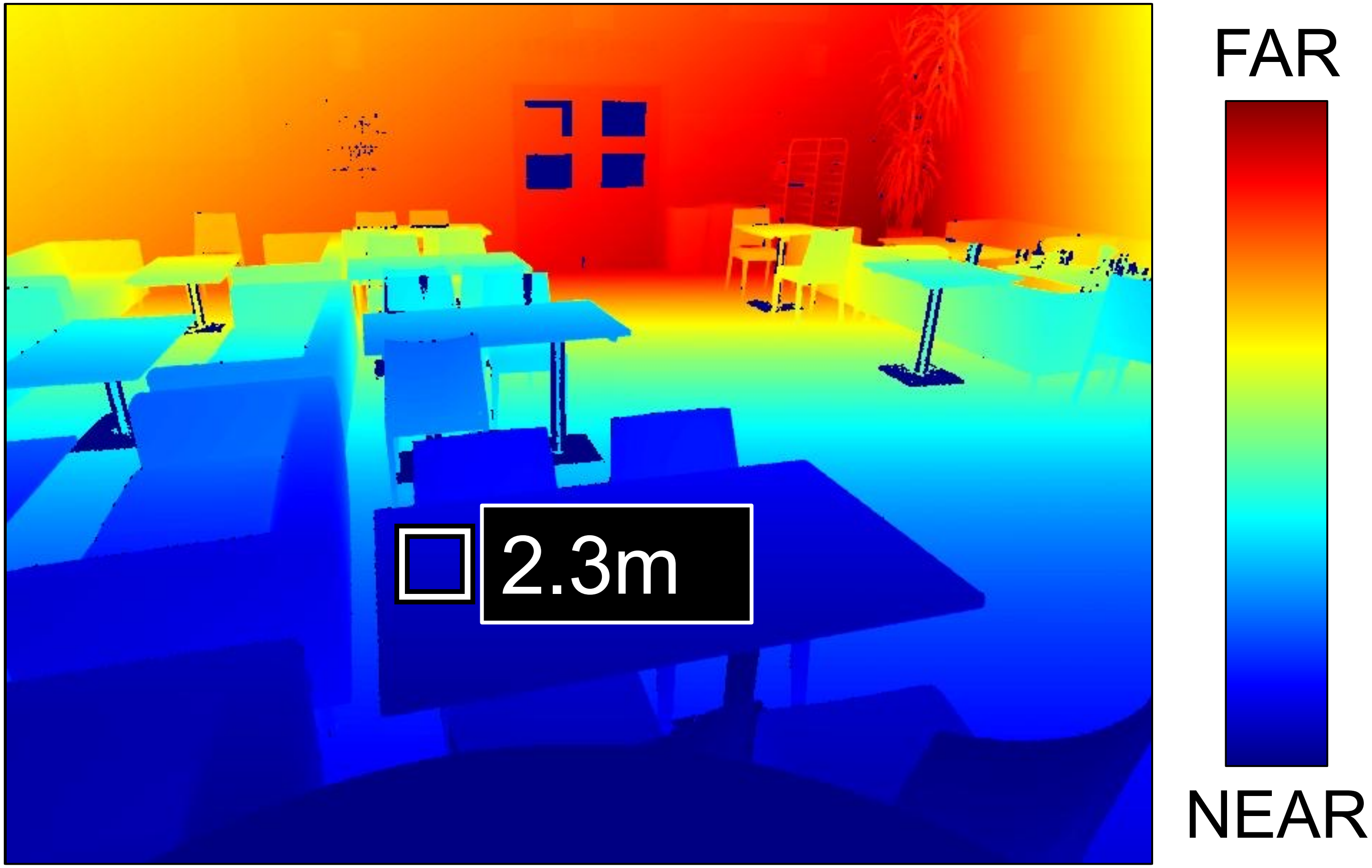
Why Are There So Many?

- Each serves different functions
  - RGB: sort of intuitive, standard, everywhere
  - HSV: good for picking, fast to compute
  - YCbCr/YUV: fast to compute, compresses well
  - Lab: the right(?) thing to do, but “slow” to compute
- Pick based on what you need and don’t sweat it: color really isn’t crucial

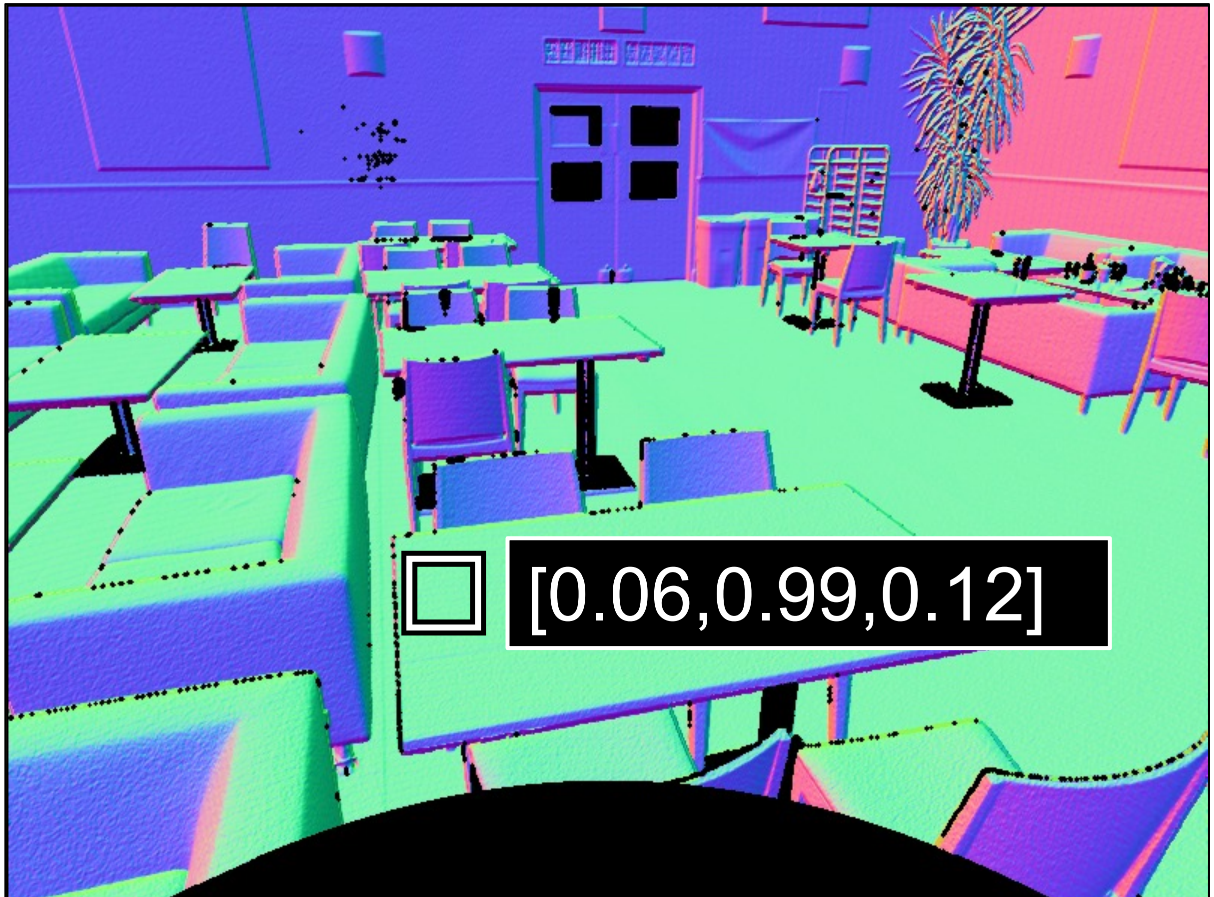
Only Images

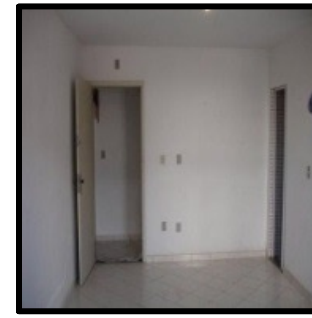
- Almost all of this class is about ordinary RGB images because this has driven a lot of applications
- However, there are lots of other images

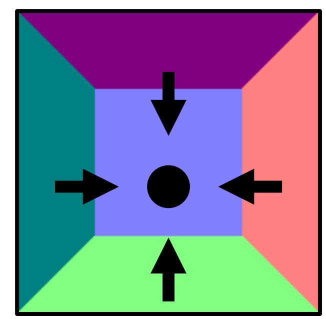
Depthmap



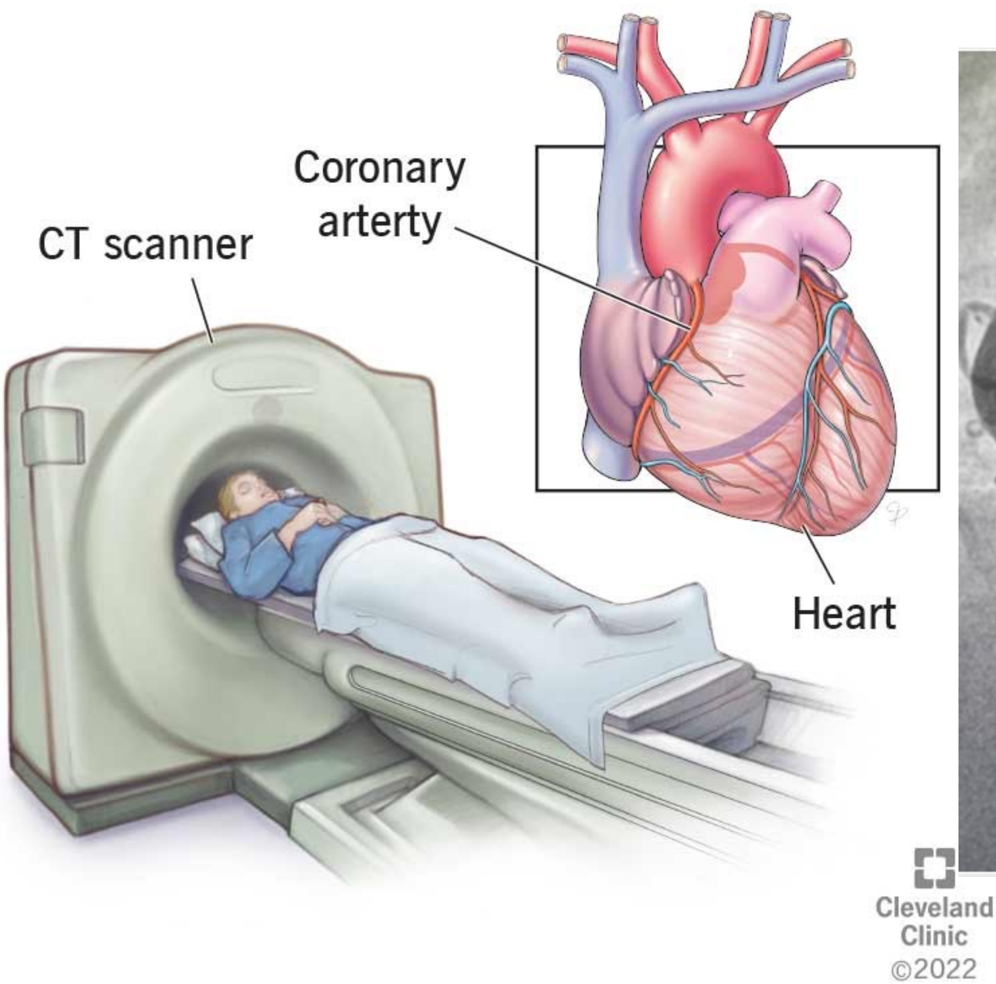
Surface Normals




Room


Legend

CT Angiogram



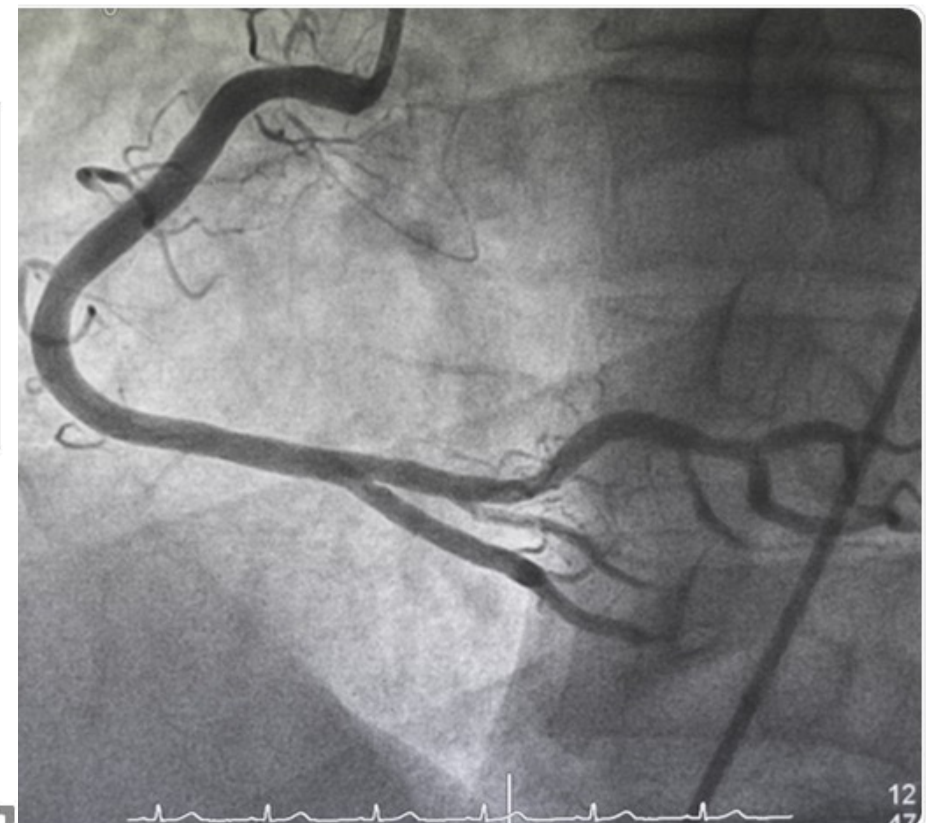


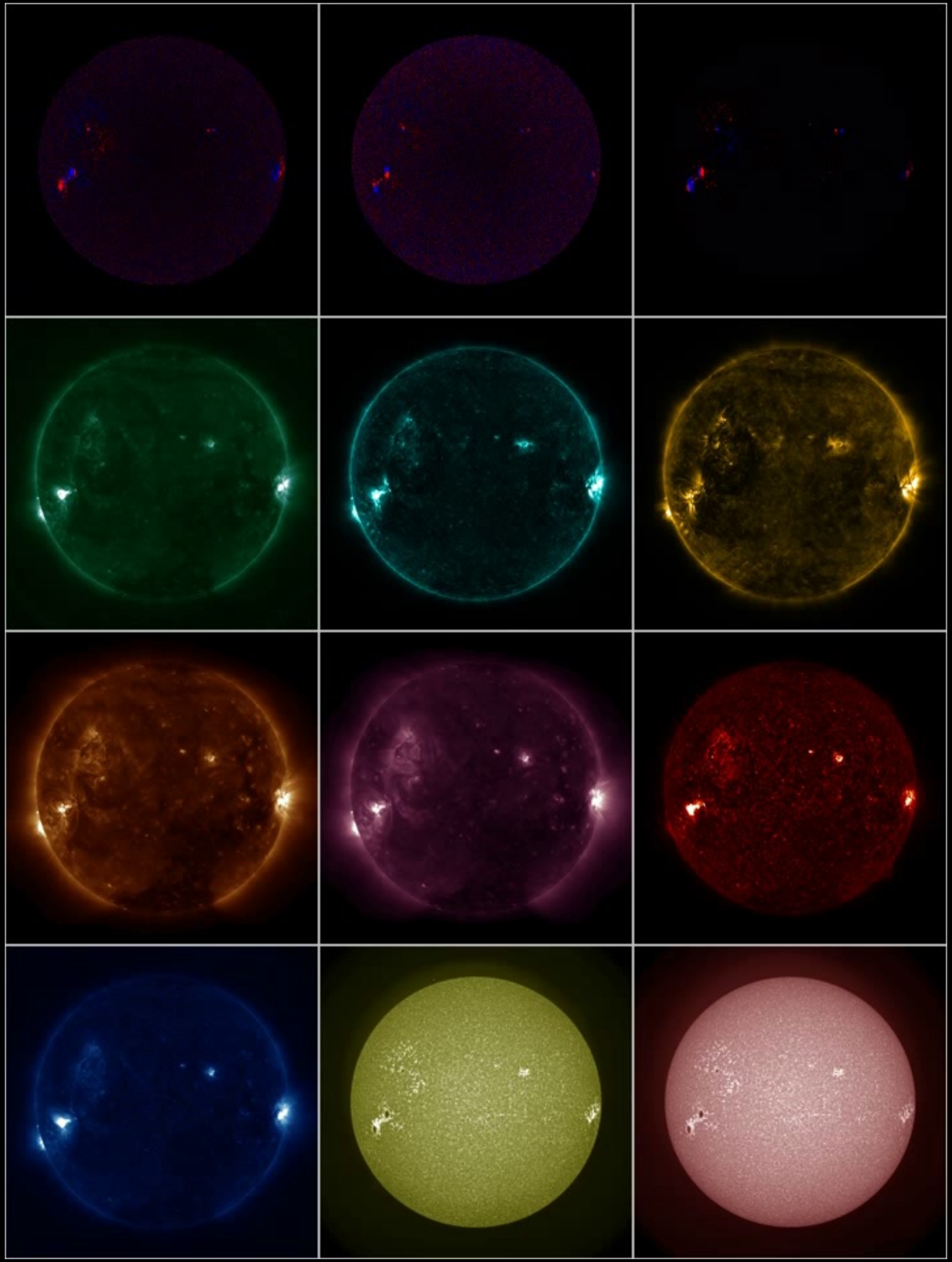
Image Credit: Clevland Clinic, New-medical.net

Science Data

Magnetic Field in:
x, y, z
via polarized light

Light at 9 ~wavelenths:
9.4nm, 13.1nm, 17.1nm
19.3nm, 21.1nm, 30.4nm
33.5nm, 160nm, 170nm

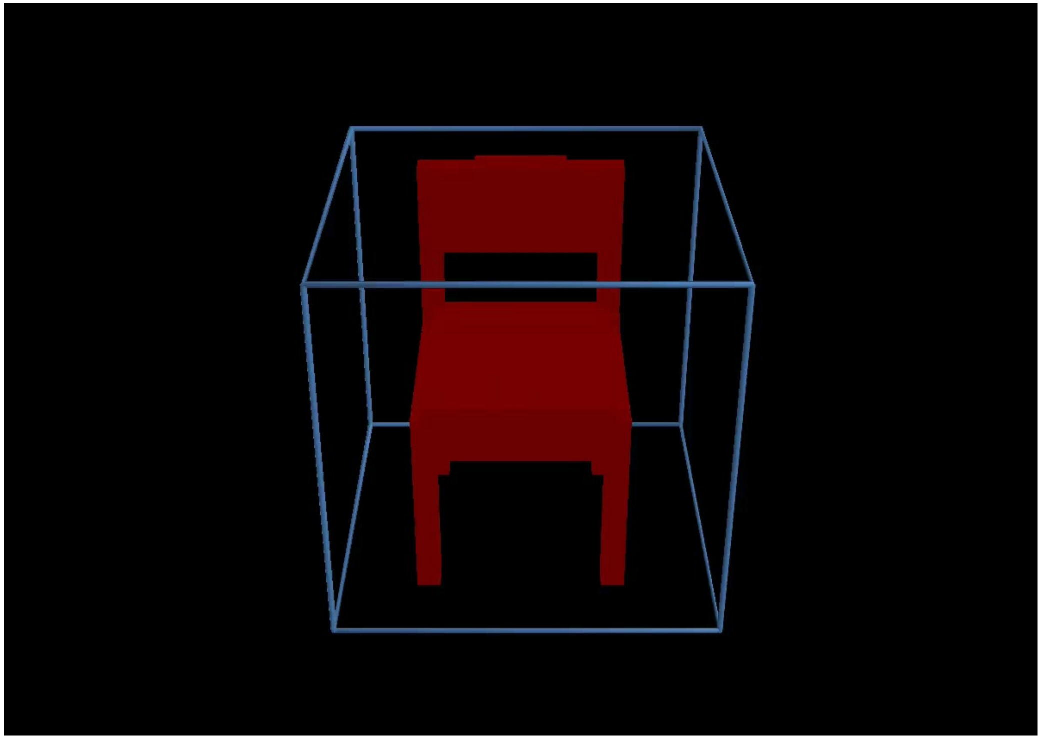
NASA Solar Dynamics
Observatory observing solar flare



Volumes

Volumes: images with more dimensions.

Emerge in 3D reconstruction, medical imaging, temporal data



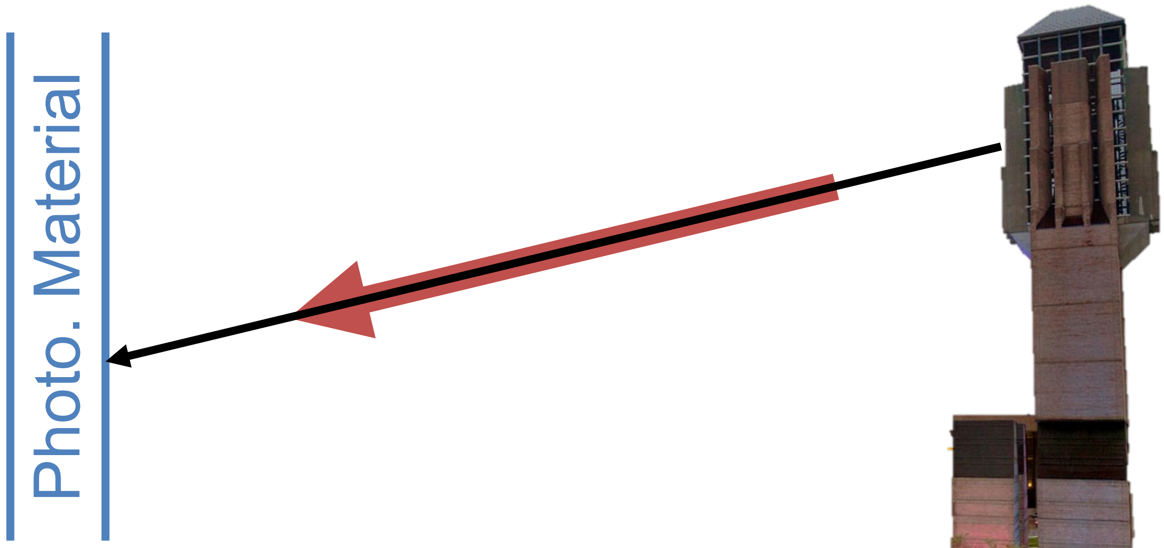
From: Girdhar et al., *Learning a predictable and generative vector representation for objects*. ECCV 2016

Other Images

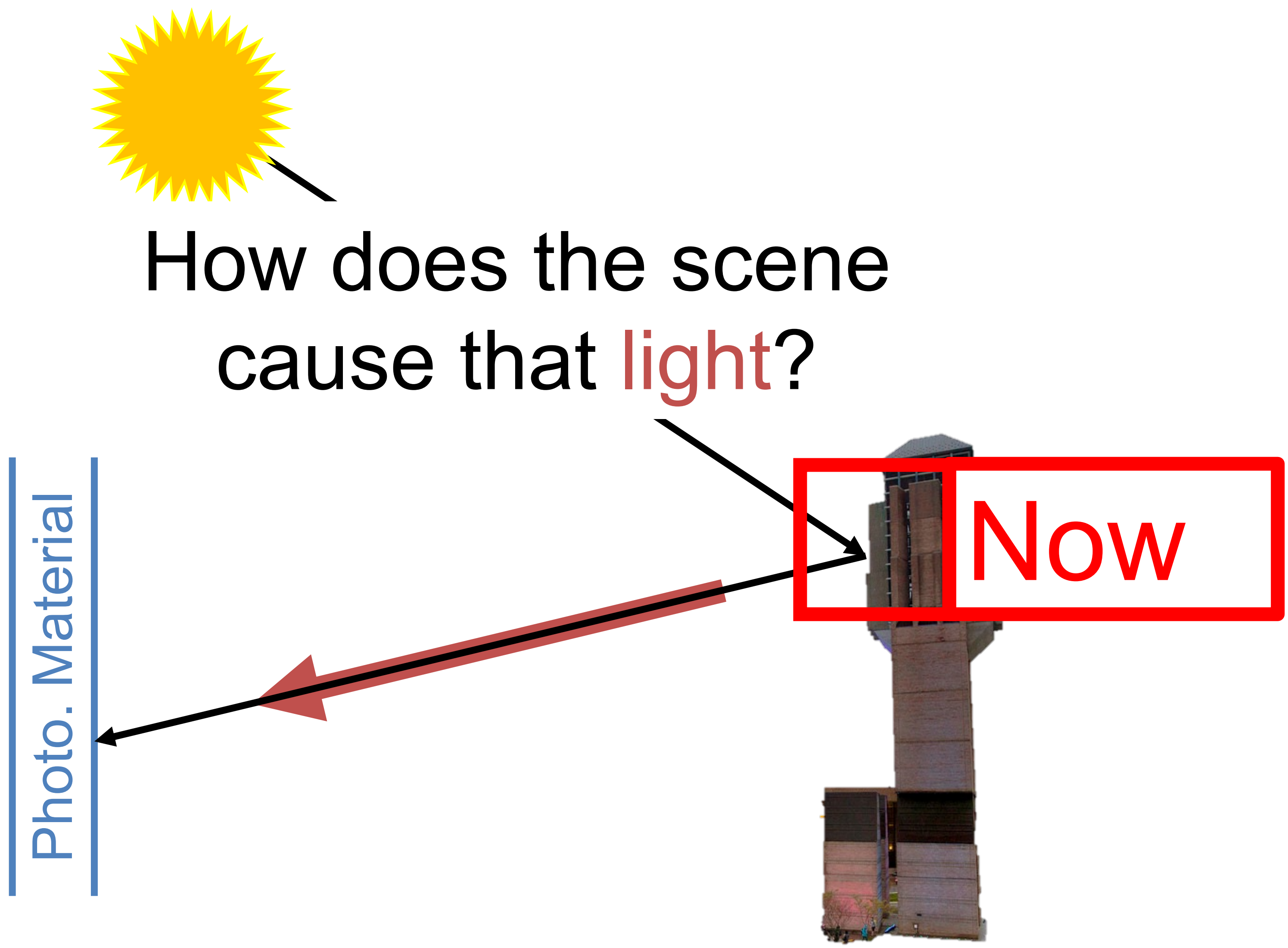
- A small part of computer vision in this class is really only for ordinary images
- The rest is easily generalized to other images
- Really transformative stuff will happen when good vision techniques get traction in other areas

So Far

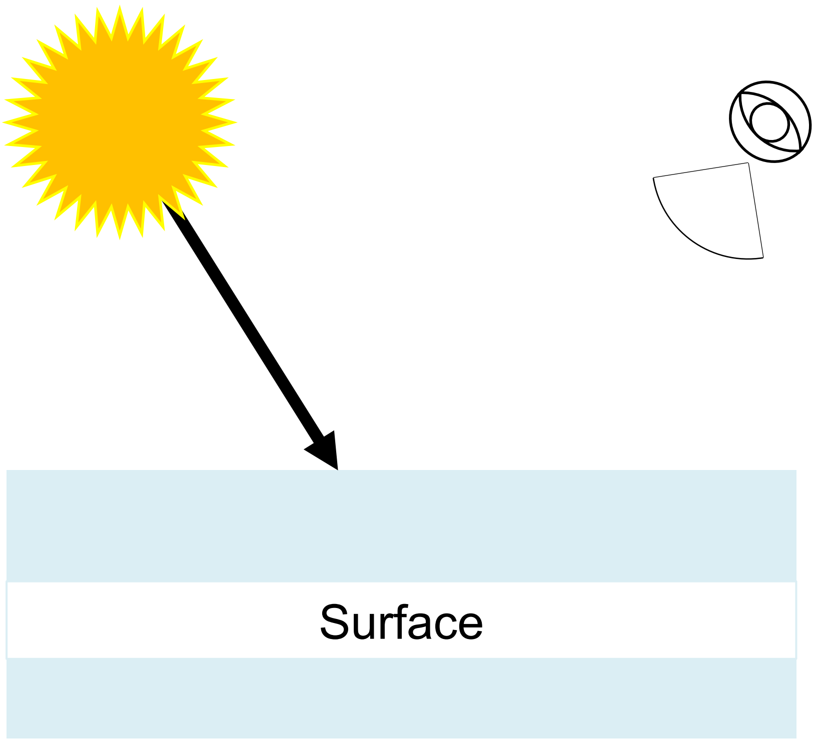
How do we represent light and its storage on film?



Now

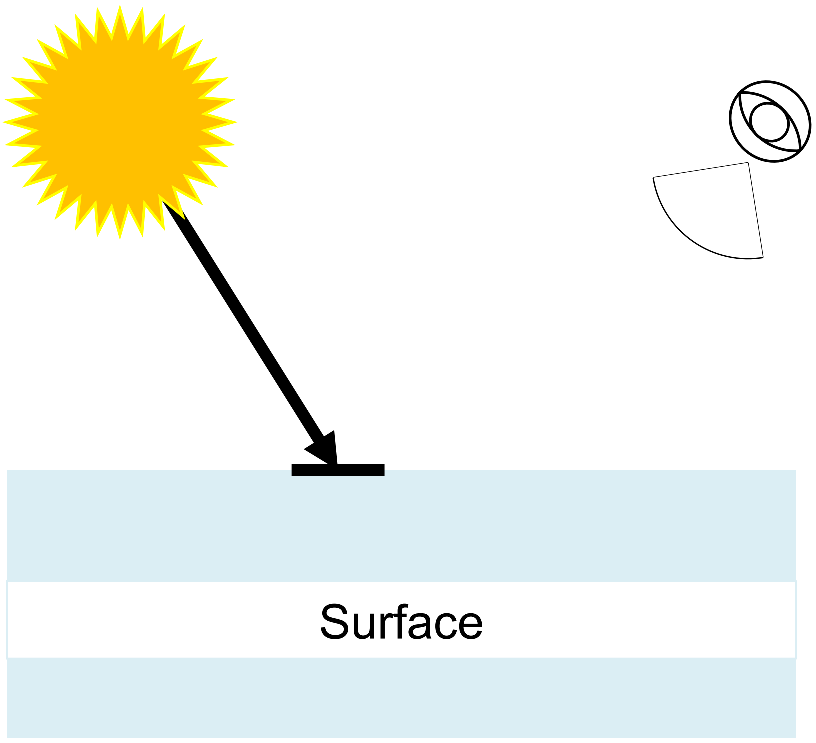


Light and Surfaces



What happens when light hits a surface?

Light and Surfaces

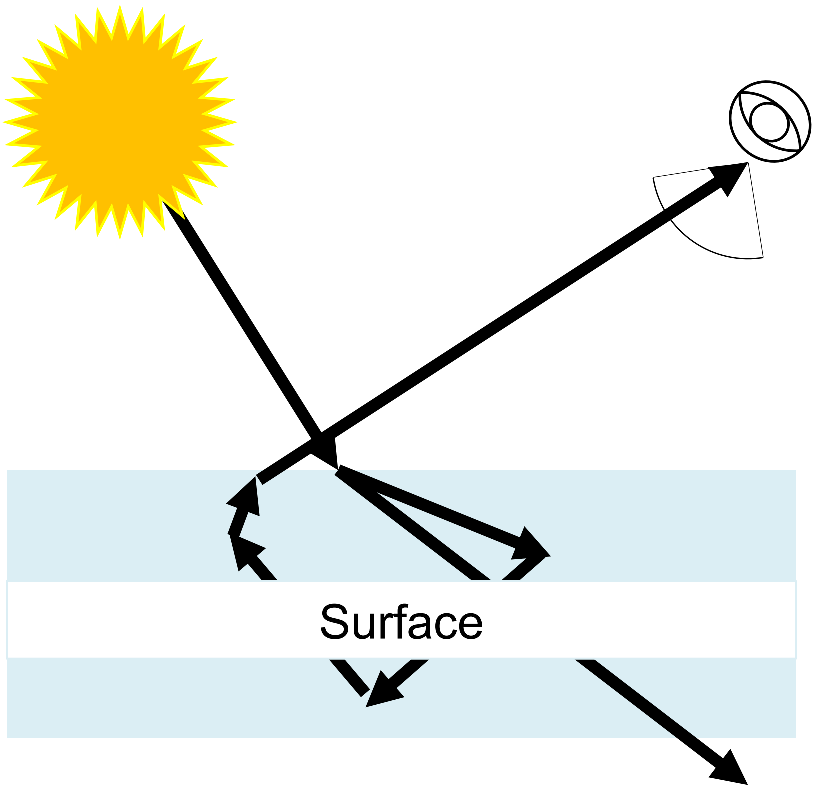


What happens when light hits a surface?

1. Absorbed

It's absorbed and converted into some other form of energy (e.g., a black shirt getting hot in the sun)

Light and Surfaces

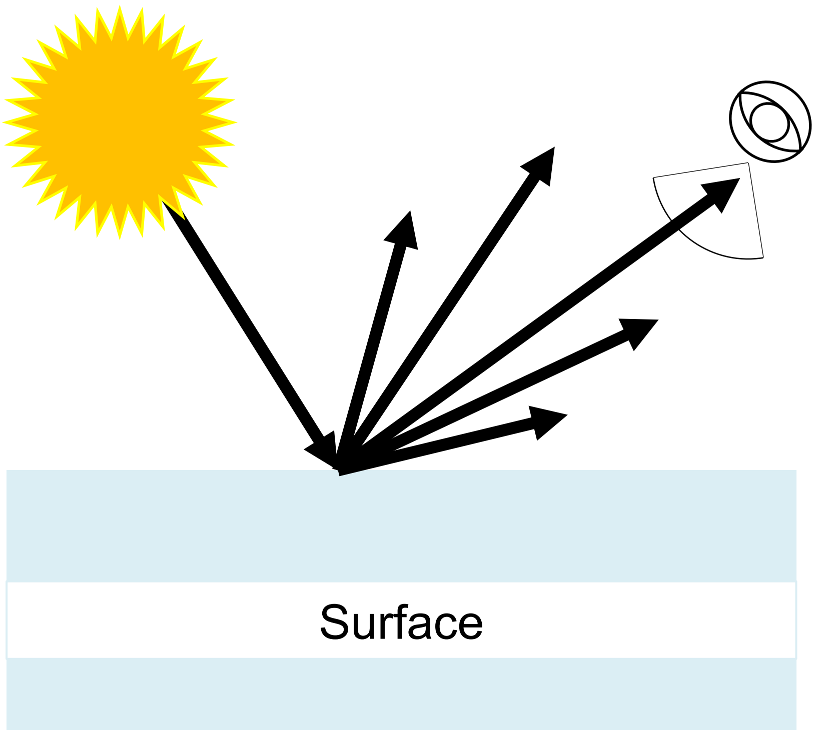


What happens when light hits a surface?

2. Transmitted

Possibly bouncing around before going through or out (e.g. lenses bend and go through, milk bounces around)

Light and Surfaces

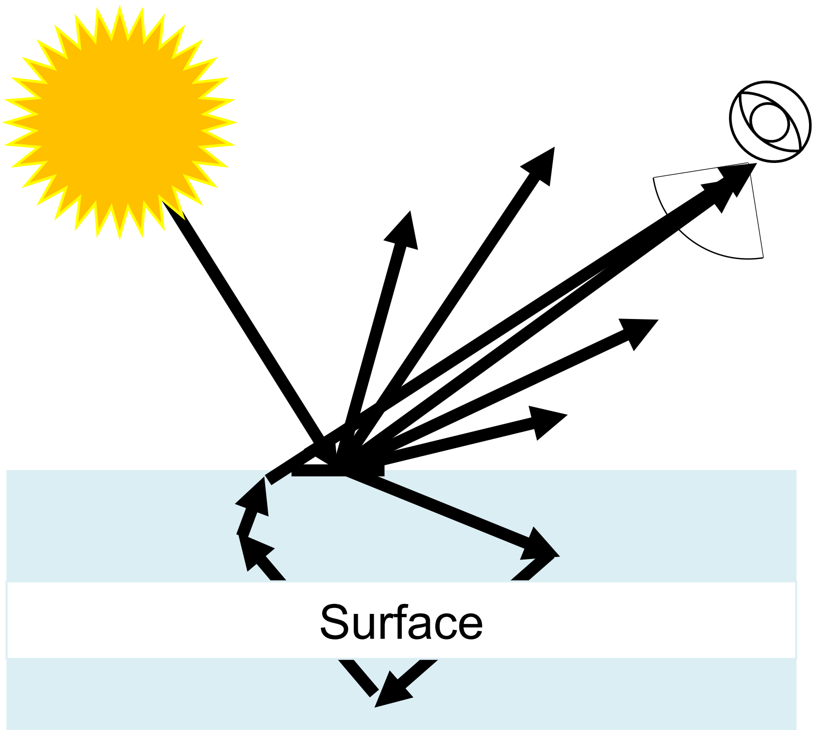


What happens when light hits a surface?

3. Reflected

It's reflected back, in one or more directions with varying amounts (e.g., mirror, or a white surface)

Light and Surfaces

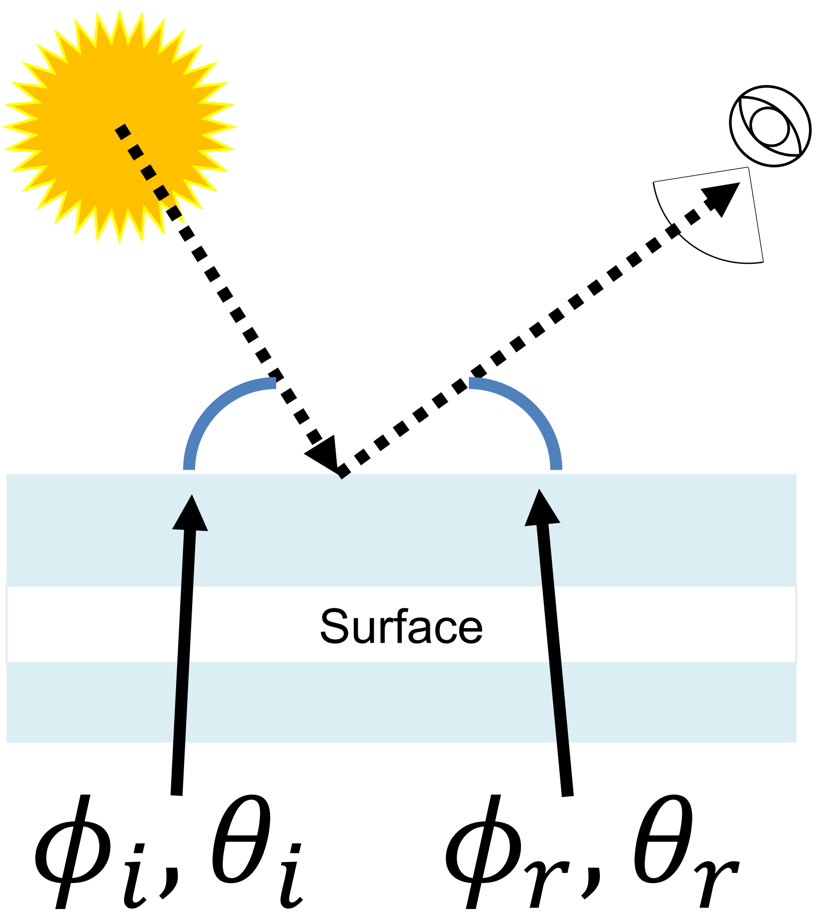


What happens when light hits a surface?

4. Everything

All of the above! Real surfaces often have combinations of all of these options.

Modeling Light and Surfaces



Opaque Reflections

Bi-directional reflectance function (BRDF): % reflected given ***i***_ncident angle_ to light ***r***_eflected angle_ to the viewer. Distribution!

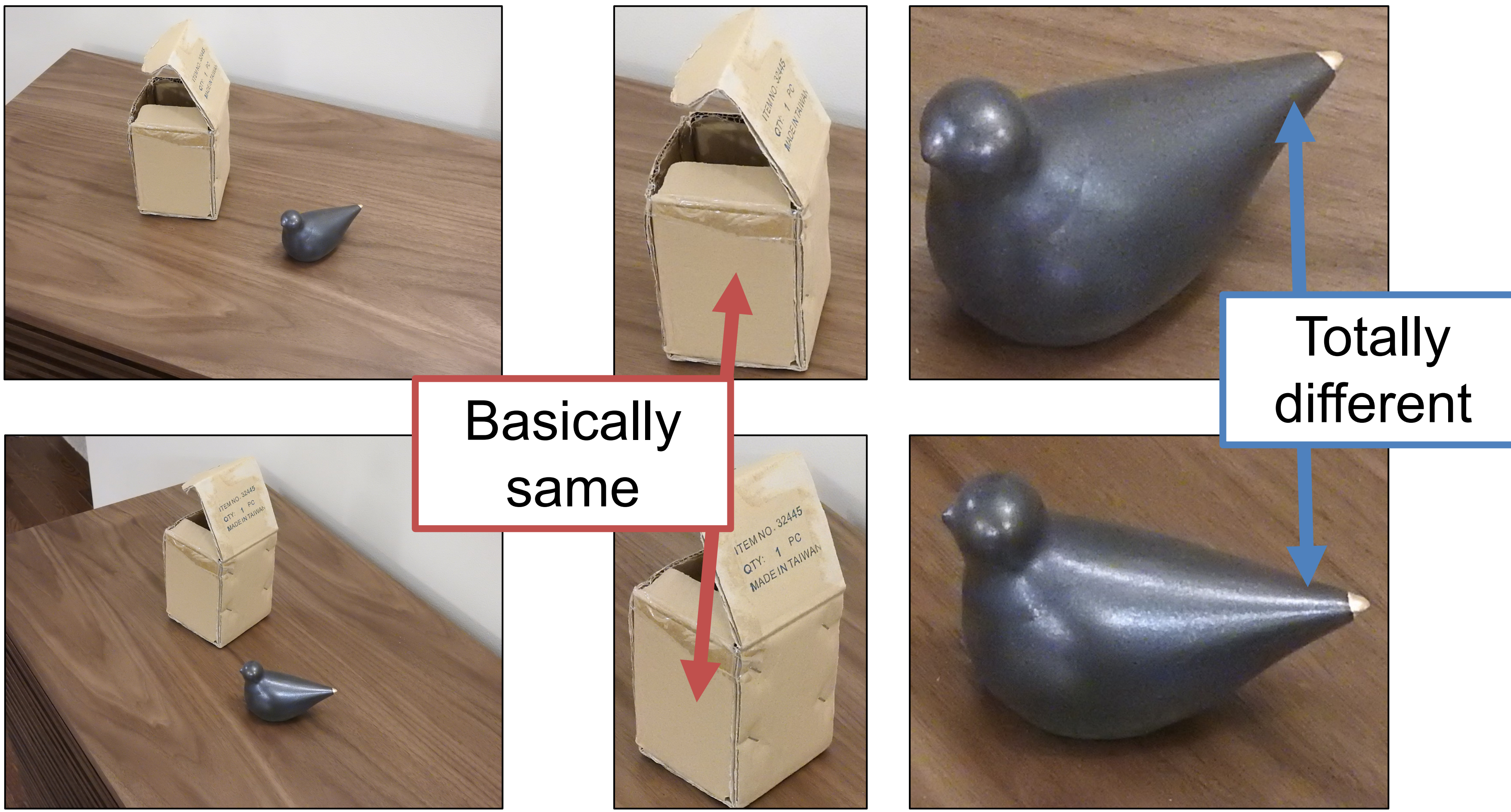
Note: have not specified form of function.

Specular and Diffuse Reflection

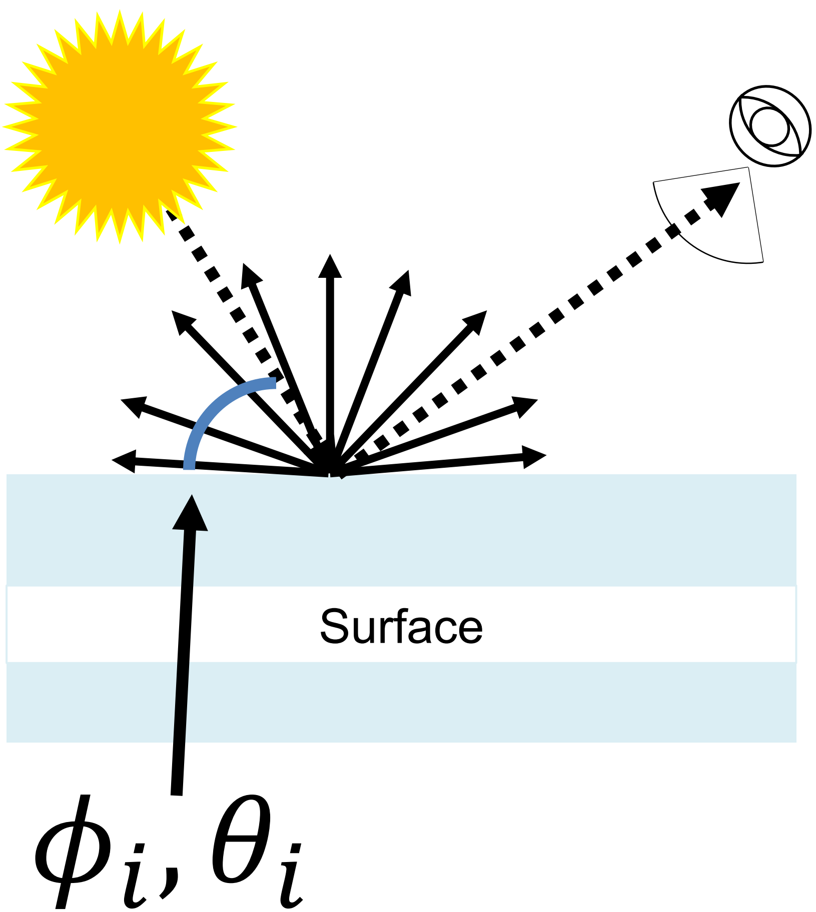
Same lighting, as close as possible camera settings, but different **location**

Specular and Diffuse Reflection

Diffuse Specular



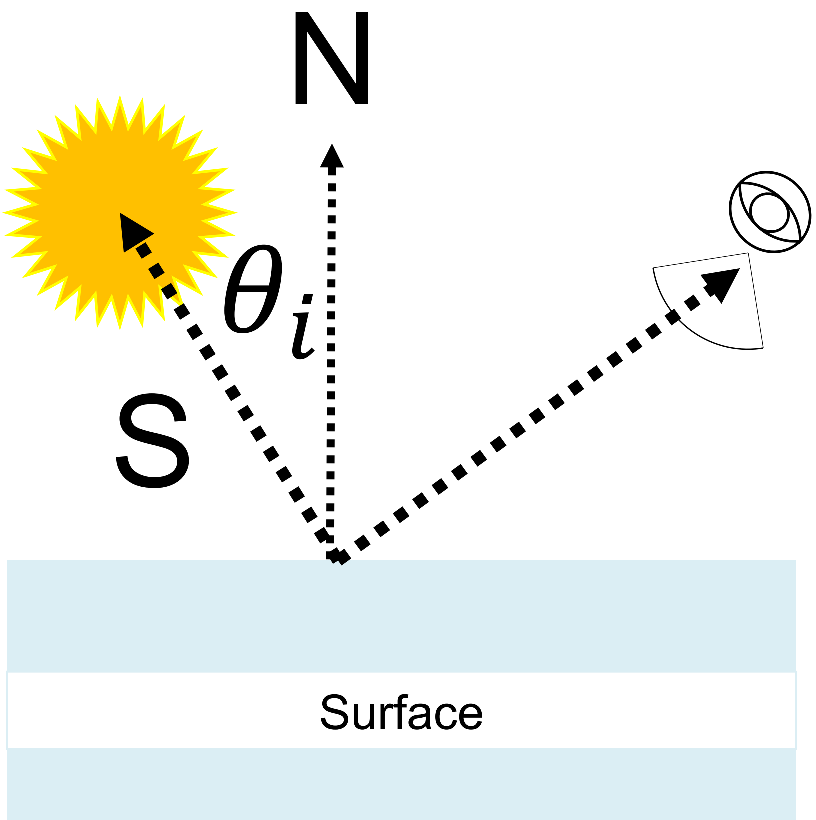
Diffuse Reflection



Lambertian Surface

Light depends **only** on orientation of surface ϕ_i, θ_i to light. Result of random small facets. Looks identical at all views.

Diffuse Reflection



Lambert's Law

N: surface normal

S: source direction **and** strength

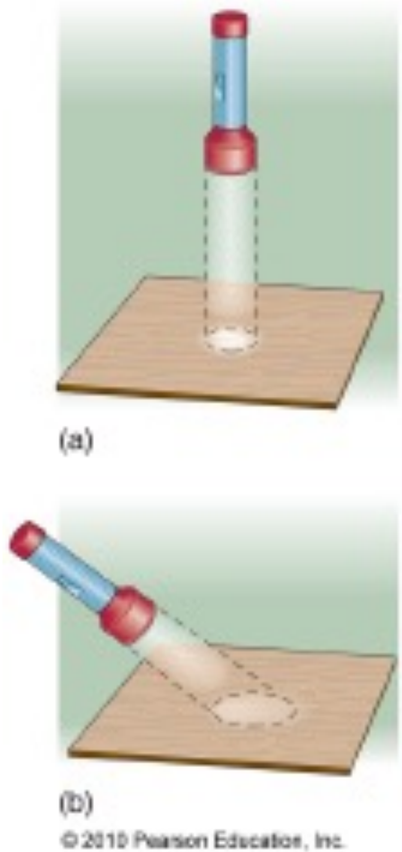
ρ: how much is reflected

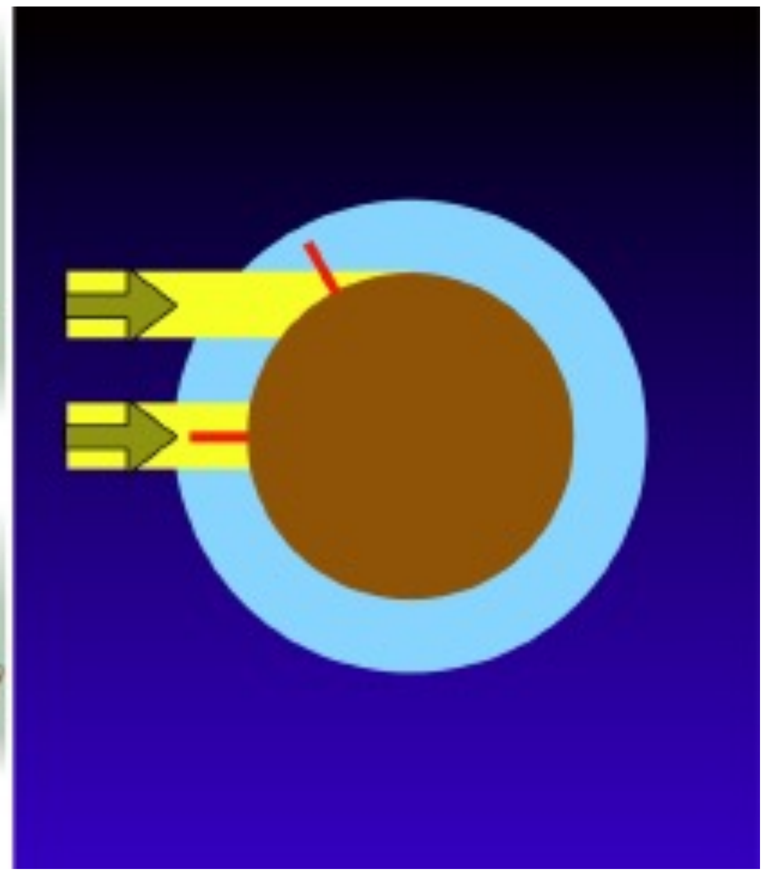
$$B = \rho \boldsymbol{N} \cdot \boldsymbol{S}$$

$$\underbrace{B}_{\text{energy}} = \rho \|\boldsymbol{S}\| \cos(\theta)$$

Lambert’s Law and Seasons

- Tilted light spreads over larger area: less energy per unit area





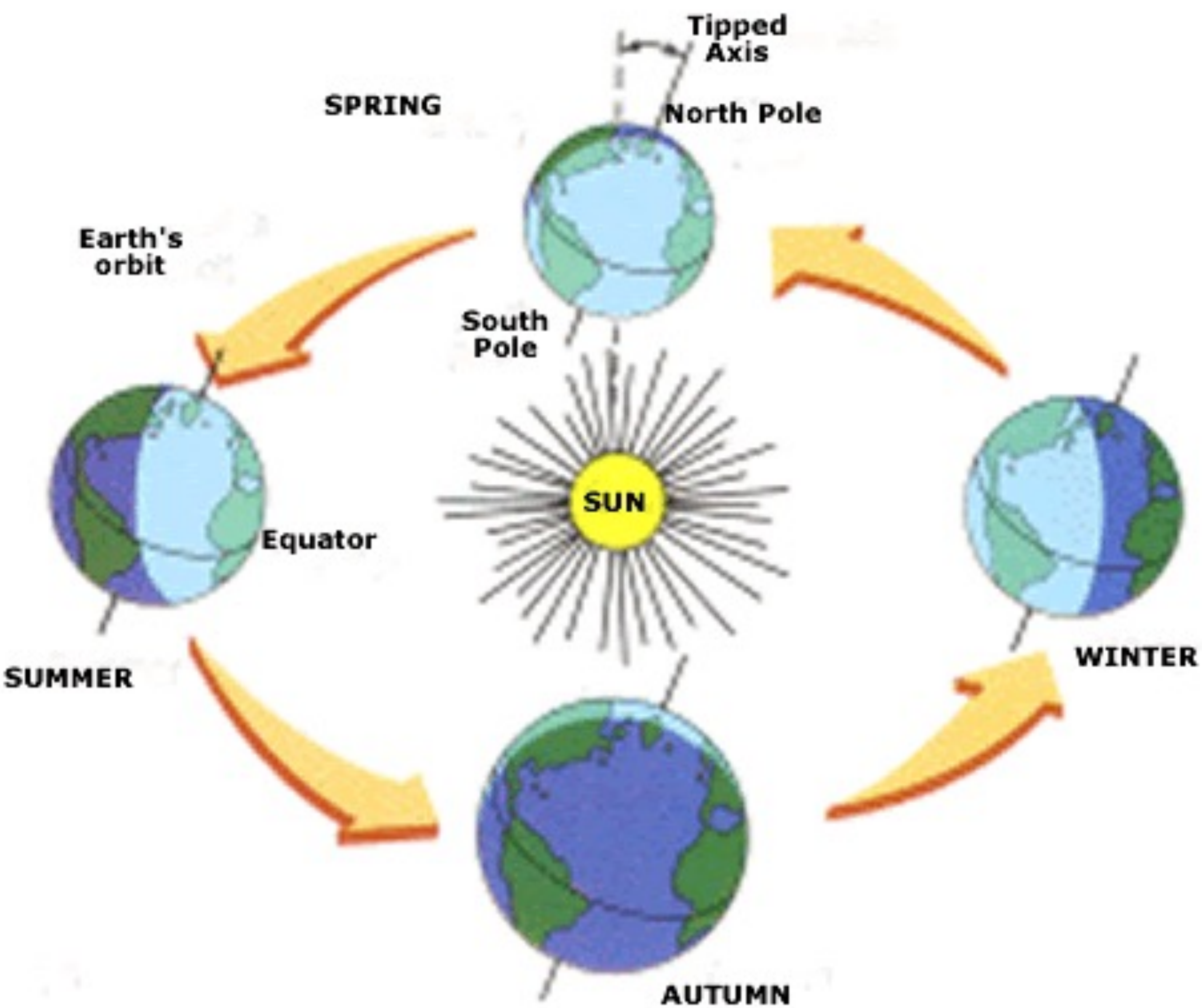
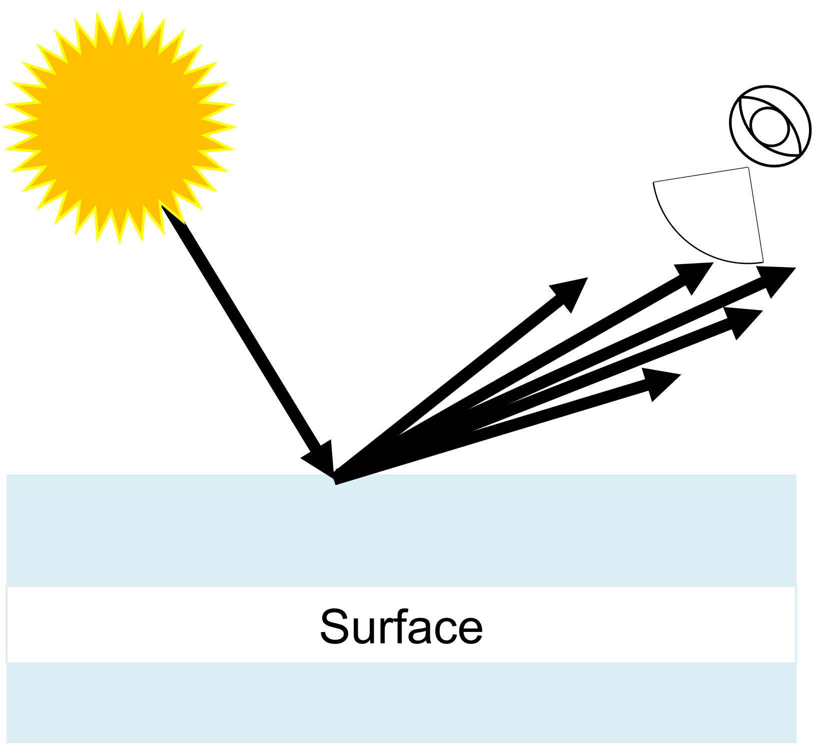


Image Credit: Quora

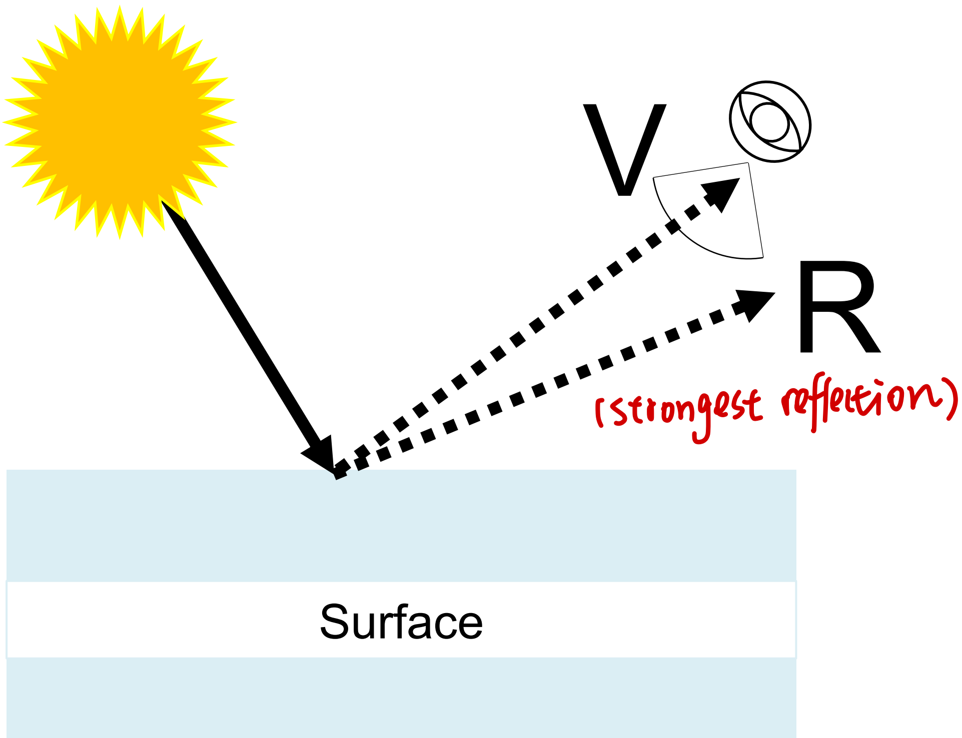
Specular Reflection



Specular Surface

Light reflected like a mirror, but spreads out in a “lobe” around the reflection ray

Specular Reflection



Phong Model

V: vector to viewer

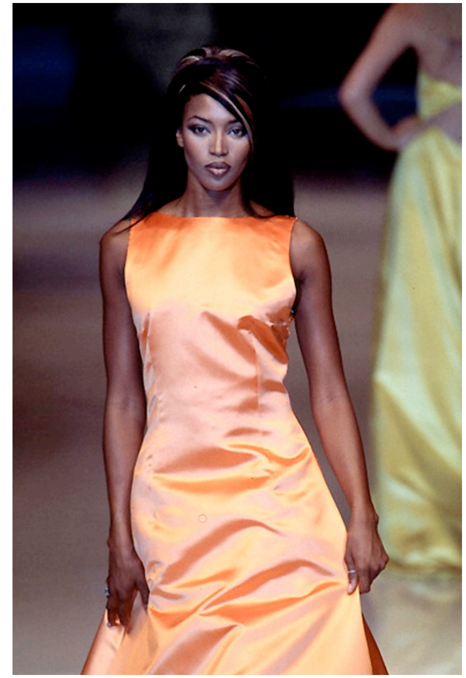
R: reflection ray

α: shininess constant

$$B = (V^T R)^\alpha$$

BRDFs can be incredibly complicated…








Slide Credit: L. Lazebnik

What Can This Be Used For

Shape from Shading

Lambert's Law: for each i of K pixels,

$$B_i = \rho \boldsymbol{N}_i \cdot \boldsymbol{S}$$

Reflected Light (1 dim) (scalar)	Surface Orientation (3? dim)	Illumination Global (3 dim)

Given: illumination & light & ρ, recover normals

Potential problems?

Shape From Shading

2D vector, Because it's a direction
and we have unit vector

$$B_i = \rho \boldsymbol{N}_i \cdot \boldsymbol{S}$$

1D, **fixed**

1D, **fixed**

actually 2D **unknown**

3D, **fixed**

- System of K equations that's underdetermined (K equations, 2K unknowns)
- **Solution**: Add more equations that enforce smoothness or finding a single surface.

Realistic Shape From Shading

$$B_i = \rho \boldsymbol{N}_i \cdot \boldsymbol{S}$$

1D, **fixed**

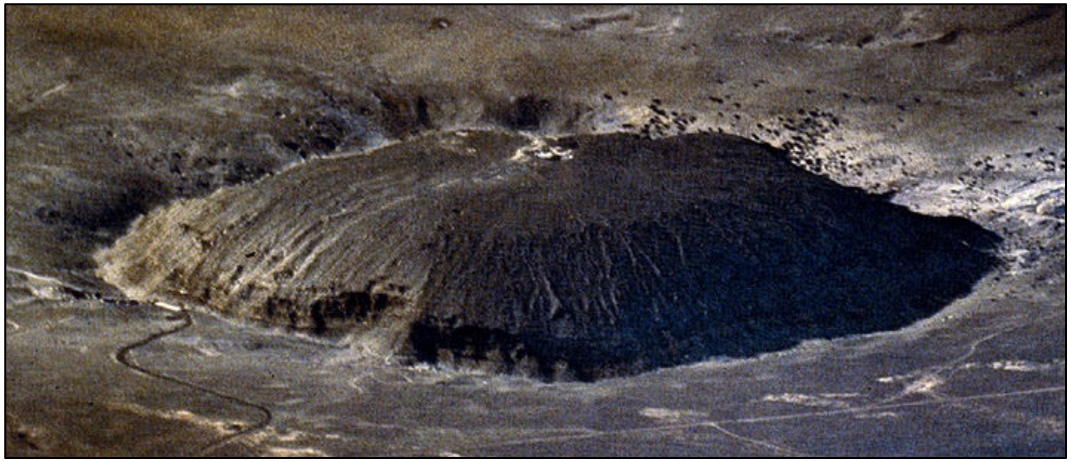
1D **unknown**

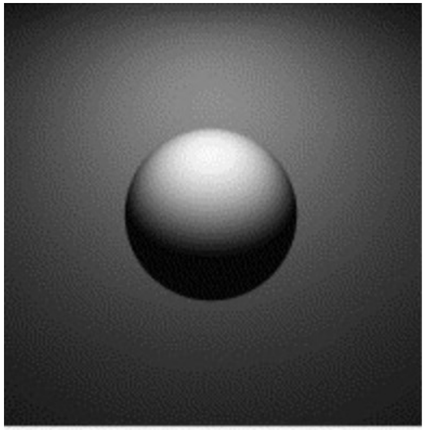
2D **unknown**

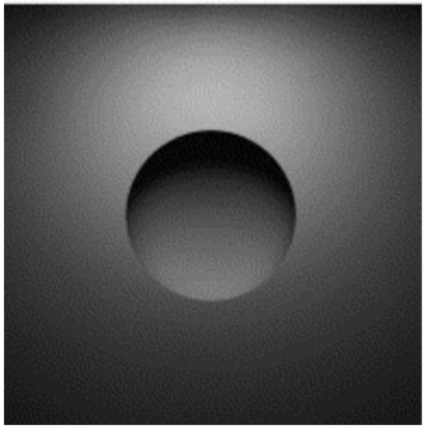
3D, **unknown**

- System of equations that's underdetermined (K equations, 3K+3 unknowns)
- **Solution**: need prior beliefs to disambiguate.

Ambiguity







Concave-Convex Ambiguity

Humans assume light from above (and the blueness also tells you distance)

Photo Credit: https://en.wikipedia.org/wiki/Meteor_Crater

Shape from Shading in Practice

https://www.youtube.com/watch?v=4GiLAOtjHNo